DOCUMENT RESUME

ED 240 706 EA 016 473

AUTHOR Moursund, David
TITLE Introduction to Computers in Education for Elementary and Middle School Teachers.
INSTITUTION International Council for Computers in Education, Eugene, Oreg.
PUB DATE 83
NOTE 97p.
AVAILABLE FROM Publications, International Council for Computers in Education, 1787 Agate Street, University of Oregon, Eugene, OR 97403 ($7.00 prepaid; quantity discounts; on non-prepaid orders, add $2.50 postage and handling).
PUB TYPE Guides - Non-Classroom Use (055)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Autoinstructional Aids; Calculators; *Computer Assisted Instruction; *Computer Literacy; Computer Managed Instruction; Computer Oriented Programs; Computer Programs; *Computers; *Computer Science; Educational Technology; Elementary Education; *Guides; Input Output Devices; Inservice Teacher Education; Introductory Courses; Microcomputers; Middle Schools; Problem Solving; *Teachers

ABSTRACT

Designed to help elementary and middle school teachers increase their level of computer-education literacy, this book discusses the capabilities, limitations, applications, and possible impact of computers in education. Chapter 1 briefly defines what a computer is and explains the book's goals and applications. Chapter 2 provides a technical overview of microcomputers and computer software and hardware, an appendix on calculators, and discussions of computer output devices, the computer's central processing unit, computer memory, and videodisks. Following chapter 3's demonstrations of how computers can provide "automatic flashcards" for students, chapter 4 describes a variety of educational computer games. In chapter 5, the author reviews the histroy of automated symbol manipulation and the development of computer languages such as FORTRAN and BASIC. Chapter 6 focuses on the use of computers for problem solving and outlines several possible applications. Chapter 7's discussion of computer and information science includes attention to modeling and simulation, information retrieval, computer graphics, artificial intelligence, and computer science's educational implications. Chapter 8 is devoted to the future of computers in business, industry, and education. The book concludes with an appendix on precollege computer literacy. (JBM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

D. Moursund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

David Moursund, the author of this book, has been teaching and writing in the field of computers in education for the past sixteen years. He is a professor at the University of Oregon, holding appointments in the Department of Computer and Information Science and in the Department of Curriculum and Instruction.

Dr. Moursund's accomplishments and current involvement in the field of computers in education include:

- Author or co-author of a dozen books and numerous articles.
- Chairman of the University of Oregon's Computer Science Department, 1969-1975.
- Chairman of the Association for Computing Machinery's Elementary and Secondary Schools Subcommittee, 1978-1982.
- President of the International Council for Computers in Education and Editor-in-Chief of *The Computing Teacher.*

This book is published by the International Council for Computers in Education, a non-profit, tax-exempt professional organization. ICCE is dedicated to improving educational uses of computers and to helping both students and teachers become more computer literate. ICCE publishes *The Computing Teacher,* a journal for teachers and for teachers of teachers. It also publishes over ten booklets of interest to educators.

QUANTITY	PRICE (U.S. FUNDS)
1-9 copies	$7.00 each
10-24 copies	$6.30 each
25-79 copies	$5.60 each
80-199 copies	$4.90 each
200+ copies	$4.20 each

This price includes shipping on prepaid orders. (A $2.50 shipping and handling charge will be added to orders that are not prepaid.)

Please place your orders with:

International Council for Computers in Education
135 Education
University of Oregon
Eugene, Oregon 97403

INTRODUCTION TO COMPUTERS IN EDUCATION FOR ELEMENTARY AND MIDDLE SCHOOL TEACHERS

Third Printing, April 1983

David Moursund
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403

CONTENTS

Preface 1
Chapter One—Introduction 3
Chapter Two—Technical Overview 9
Chapter Three—Automatic Flashcards 23
Chapter Four—Educational Games Computers Play 33
Chapter Five—Automated Symbol Manipulation 43
Chapter Six—Problem Solving in a Computer Environment 51
Chapter Seven—Computer and Information Science 61
Chapter Eight—The Future 71
Appendix A—Calculators 79
Appendix B — Pre-College Computer Literacy: A Personal Computing Approach 83

INTRODUCTION TO COMPUTERS IN EDUCATION FOR ELEMENTARY AND MIDDLE SCHOOL TEACHERS

PREFACE

Public education has the reputation of being rather conservative and slow to change. Certainly, education is a massive institution, beset by many problems. Still, one can point to major changes that have occurred in the past and to others that are currently going on.

Looking back to ancient history, there was a time when reading and writing did not exist, and when schools or formal education did not exist. Clearly, we have come a long way since then! In more recent history, one can look at the invention of the printing press and movable type as major turning points. It has become feasible to think of universal literacy. It is hard to grasp the impact this idea has had upon society.

Calculators, computers, videodisks and related electronic technology are now becoming readily available. These have the potential to impact education at a level perhaps comparable to that of the printing press. As an educator, you are fortunate to be living at the time of a major change and able to participate in it. Such change is exciting and challenging.

It is likely that you own a calculator and have a reasonable level of skill in its use. The typical adult can learn to perform the four basic functions on a calculator with a few minutes of practice and instruction. Of course, there is much more to learning to use a calculator as an effective aid to problem solving than can be covered in a few minutes. Entire books have been written on this topic, including two by your author.

Computers have a similar characteristic. It takes only a few minutes of instruction and practice to learn to use a computer. Running computer programs that are stored in a program library is roughly as difficult as playing a record on a stereo system. Very young children have no difficulty learning to accomplish this task. As with calculators, however, there is much more to the computer field than this. Educators need to have a substantially deeper insight into computers in education than can be gained in a few minutes of study.

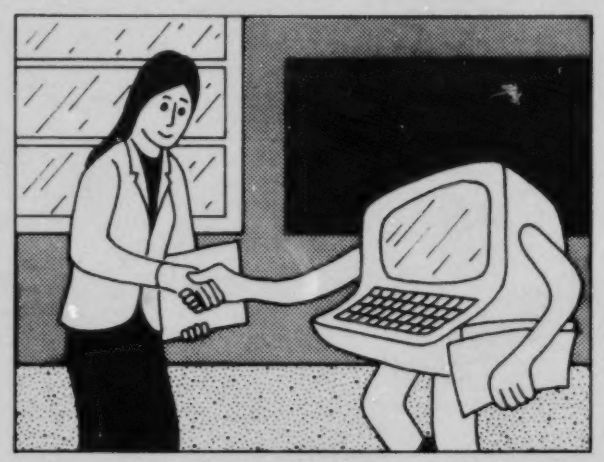
There is no magical shortcut to gaining a serious and deep understanding of computers. A teacher needs to know something about computer hardware—its capabilities, limitations and price range. A teacher needs to know about computer software—the availability, quality and costs. A teacher needs to know how computers can impact both the process and content of education. A working knowledge of all of these things, suited to the demands of a modern, computer-equipped school, is called computer-education literacy.

This book is designed to help increase your level of computer-education literacy. It can be used in a formal course for inservice or preservice teachers, or it can be used for self-instruction. In any case, you should realize that there are two major components to computer-education literacy.

One component is computer programming—learning to develop computer programs and materials useful to you and your students. There are literally hundreds of computer programming books on the market, with many new ones appearing each year. There are also about a half dozen widely used programming languages, and many more that see limited service. Thus, this book does not attempt to teach computer programming, even though it does contain a chapter on the topic. It is likely and quite desirable that a formal computers-in-education course taught using this book will contain instruction in computer programming. Many of the ideas discussed here will be more clearly understood by trying them out on a computer, and by developing related computer programs.

This book does cover the second major component of computer-education literacy. It discusses the capabilities, limitations, applications and the possible impact of computers in education. It contains a broad general overview of hardware, software, courseware and computer-related information that teachers need to know.

This book also contains many applications of immediate use—things that you can do to help your students learn about computers, and to help prepare them for life in a computerized society. Many of the applications do not require access to computer equipment; those that do are suitable for use by a teacher who is just learning about computers.



Ideally, the person studying this book will also be receiving instruction in programming a computer and will have the opportunity to gain substantial hands-on experience in using computers. However, this book has been designed to be useful to teachers who have little or no access to computers at the current time. If that is your situation, then it is suggested that you find someone else who is also interested in studying this book. The two of you can work together and discuss many of the ideas covered in the book. This will add to your enjoyment of the material and will greatly enhance your understanding of it.

If you have a choice of languages to study while using this book, give first consideration to Logo. It is an excellent first language for students and teachers, and it is particularly relevant to the elementary school level. If Logo is not available, then BASIC is likely to be your choice. If possible, gain some experience with PILOT, also.

This book contains two Appendices. The first is a brief summary of ideas that are covered in a calculators in education course for elementary and middle school teachers. The second is a detailed discussion of what is meant by computer literacy for students. If you already have some background in educational uses of computers, you should read Appendix B before starting the earlier chapters. If you are just beginning your computer studies, wait until you have made some progress in the field. In either case, plan to read Appendix B a second time as you near the end of the eight chapters. It provides a focus on how to apply the computer knowledge you are gaining.

The author recognizes that the United States is committed to conversion to the metric system of measurements. Thus, many of the exercises, activities and discussions in this book make use of metric measurement. However, some of the standards in the United States computer industry are not stated in metric measurement. A commonly used reel of magnetic tape is 2400 *feet* in length and recorded at a density of 1600 characters per *inch*. Because of this, the reader will find a mixture of the two systems of measurement used in this book.

CHAPTER ONE

INTRODUCTION

As you walk into a third grade classroom, you see a student sitting in front of a TV set with a typewriter keyboard. You look over the student's shoulder and read:

HELLO. THIS IS A "NUMBER FACTS" DRILL PROGRAM. IF YOU WANT TO PRACTICE ON NUMBER FACTS PUSH THE ENTER KEY.
?

$3+7=10$
$4\times3=12$
$\frac{16}{4}=4$

You notice that different number fact sentences keep appearing and disappearing on the bottom half of the screen.

The student, obviously familiar with this program and the equipment, pushes the ENTER key almost before you finish reading the directions. The printing on the TV screen disappears, but is immediately replaced.

GOOD, I AM PLEASED THAT YOU WANT TO PRACTICE YOUR NUMBER FACT SKILLS. PLEASE TYPE IN YOUR FIRST AND LAST NAME, AND THEN PUSH THE ENTER KEY.
?

You watch as the student rather laboriously types in TERRY JOHNSON and pushes the ENTER key. The computer quickly responds.

OKAY TERRY, MY RECORDS SHOW YOU NEED TO WORK ON YOUR "TIMES" FACTS. IF THAT IS ALL RIGHT WITH YOU TYPE YES AND THE ENTER KEY. OTHERWISE TYPE NO AND THE ENTER KEY.
?

Terry, being an obedient and rather docile student, types YES and the ENTER key.

VERY WELL, TERRY. THIS IS A ONE MINUTE SPEED DRILL. WORK AS FAST AS YOU CAN. TRY NOT TO MAKE ANY ERRORS. AFTER YOU TYPE AN ANSWER YOU MUST PUSH THE ENTER KEY. PUSH IT NOW WHEN YOU ARE READY TO BEGIN.
?

You see Terry take several deep breaths, poise one hand over the digit keys on the keyboard and push the ENTER key. The screen clears and a multiplication problem is displayed; numbers in the left upper corner of the screen begin to count off seconds and tenths of a second.

1.4 SECONDS
1 RIGHT
0 WRONG

$$\begin{array}{r} 6 \\ \times 2 \\ \hline ? \end{array}$$

You watch as Terry completes 35 problems in a minute. Each wrong answer is accompanied by a "beep" from the computer along with a display of the problem solved correctly. At the end of a minute the computer displays the results.

> VERY GOOD TERRY! YOUR SPEED AND ACCURACY ARE BOTH BETTER THAN YESTERDAY. YOU COMPLETED 35 PROBLEMS WITH ONLY 2 ERRORS. WOULD YOU LIKE TO TRY ANOTHER MINUTE DRILL? PLEASE TYPE YES OR NO, AND THE ENTER KEY.
> ?

Does this seem like science fiction—like an adventure into the future? It isn't! This is a routine and ordinary use of computers in elementary education. A few schools have had computer facilities such as this for many years. For most, however, it is quite new or still to come. If computers are already able to make this type of contribution to education, what will schools be like 25 years from now? Will there be (human) teachers and will students come together in classrooms like they do today?

Quite likely yes, but there will also be many changes. The greatest changes will be based upon computer technology. Students will carry on conversations with computers, learn from computers and learn to direct computers in their operations. (Much of this may take place at home.) Computers will routinely solve many of the problems that current students spend days or even weeks studying, and perhaps never master. Computers will help students to learn more, better and faster. But equally important, they will change what students need to learn and what it means to "know" something.

This book is for preservice and inservice elementary and middle school teachers who want to participate in and help guide the changes that seem inevitable. It contains a broad general introduction to computers and to their educational applications. It also contains a large number of applications for the elementary and middle school classroom. Many of these are ideas you can use immediately, whether or not computers are available in your school.

The elementary and middle school teacher is a generalist, responsible for instruction in a wide variety of disciplines. A computer is a general-purpose aid to problem solving, useful in every academic field. The examples and activities discussed in this book are designed to fit the needs of a generalist—a person with multidisciplinary interest. No special, college level knowledge of any particular discipline (such as mathematics) is assumed.

Electronic Digital Computing Devices

In this book the word calculator refers to an electronic digital calculator that can add, subtract, multiply and divide. These range in price from about $6 to $600; most homes in the United States have one or more.

Similarly, the word *computer* refers to an electronic digital computer. It works with alphabetical and numerical data coded in a digital form, using a binary digital code. Electronic digital computers range in price from about $200 to $12 million or more. Hundreds of thousands of computers have been sold in the United States. They are common in secondary schools and higher education; they are rapidly becoming a common item in the elementary and middle school, and in the home.

There are hundreds of items besides calculators and computers that make use of electronic digital circuitry. Examples include most new automobiles, many microwave ovens, electronic games and many new color television sets. For the most part, we will not be discussing such devices in this book. Rather, we will focus on the electronic digital computer, occasionally discussing calculators to help illustrate computer ideas.

The Brain Extender

We are all familiar with machines that supplement and extend people's physical capabilities. Examples include the car, train and airplane; telescope and microscope; telephone and telegraph; radio and television; eye glasses and hearing aids. The electronic digital computer is a brain extender; it has the capability of supplementing and extending people's mental capabilities.

The idea of a brain extender sounds like science fiction, and perhaps is somewhat frightening. Two examples will help allay your fears. Suppose you were reading a book and came across the word timocracy. Chances are you would need to look it up in a dictionary, thereby using the dictionary as an extension of your memory and knowledge.

Next, suppose you were solving a math problem and found that you needed to know the square root of 897.48. Chances are you would key this number into a calculator and push the square root key.

Use of a dictionary is taught in our educational system. We memorize the spelling and meaning of commonly used words, and we learn to use a dictionary to look up uncommon words or words we have forgotten.

Use of a calculator, however, is not yet commonly built into our curriculum. Mathematics educators have not yet agreed upon what calculations students should memorize or learn to do mentally, which they should learn to do using pencil and paper, and when they should be allowed to use a calculator.

The problem of computers is much more complex, because the range of capabilities is so much greater. This range is continually being expanded via research, as faster, cheaper, more capable machines become available. For example, voice input to a computer is now possible, providing the vocabulary is sufficiently limited. Twenty-five years from now a voicewriter (voice driven typewriter) may be common. Education will be challenged to adjust to such technological advances.

Goals

This book has several goals. The major ones are:

- To help you cope with computers as they come into your school.
- To help you prepare your students to cope with the computerized aspects of life in school and in our society.
- To provide a foundation so you can learn more about computers in the future.

These are rather ambitious goals. Fortunately, we do not start from scratch. It is assumed you are already quite knowledgeable about education. You have insight into students, what they can learn and how they learn.

It is likely that you already know quite a bit about calculators and that you probably own one. If you have not had a formal course on calculators, it is recommended that you read Appendix A of this book. It provides a brief summary of some of the key ideas contained in a book on calculators for elementary and middle school teachers, written by this author.

Finally, whether you realize it or not, you know a lot about computers. For example, consider our telephone system. It is often likened to a gigantic computer network. The telephone system contains thousands of computers. Dozens of these may come into play in routing and billing a long distance phone call. The fact that all of this occurs automatically and very rapidly tells you that computers can be easy to use and can rapidly solve complicated problems.

It is likely that you have made use of an airline or hotel reservation system. You have received bills prepared by computer, and you have received checks printed by computer. Your income tax forms are checked by a computer, and your social security records are maintained by a computer. Thus, you know that computers are widely used in our society and that they affect most of us in our everyday lives.

Unfortunately, you may also possess some incorrect information. You have been exposed to the walking, talking robots and computers of movies and television. You may have read books about computers that attempt to take over the world and subjugate the human race. You have read newspaper and magazine articles about computers, perhaps writen by people who have little insight into the field. This incorrect information is intermingled with your correct knowledge.

This book will help you to learn what computers are currently doing, what they can do, and what they cannot do. It will correct your misconceptions as it adds to your factual knowledge. While the main emphasis is upon computers in education, much of the information is of a more general nature. Thus, this book will help you to cope with the computer-oriented aspects of life in our society.

If you haven't already done so, go back and read the Preface. It will help clarify the direction and intent of this book. Note that, ideally, a course based on this book should include substantial hands-on experience with computers. If you have not yet used a computer, now is the time to gain this initial experience. If you are enrolled in a formal course, your instructor should provide you with access to computers. If you are studying on your own, you may be able to gain some computer access via a nearby school. Alternatively, visit a store that sells computers. There are many thousands of such stores now, and they will be happy to give you a demonstration and some hands-on experience. Here, of course, you will have to listen to a sales pitch. Don't buy! Wait until you know more and can judge whether or not you really want or need to own a computer.

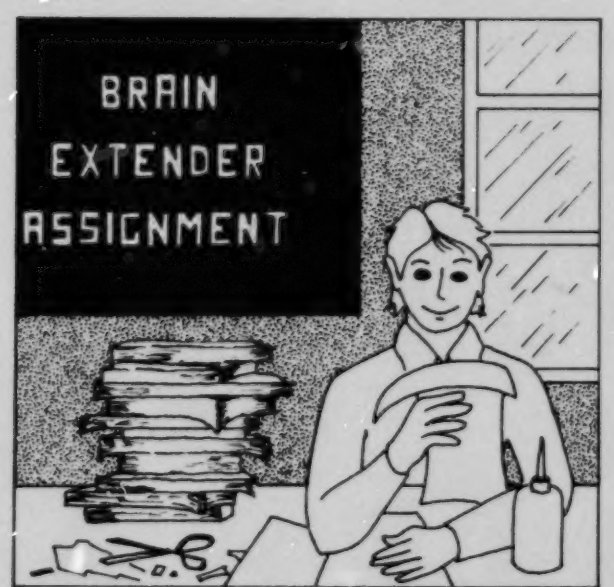
Applications

This book contains a number of sections headed Applications. These contain activities and exercises suitable for use in an elementary or middle school classroom. No attempt has been made to classify these applications by grade level; indeed, most can be used over a wide variety of grade levels. Also, you will find that most of the applications do not require use of a computer. Computers are still relatively rare in the elementary or middle school classroom. Thus, while a teacher may have access to one or two microcomputers, it is very rare that a teacher has easy access to a classroom set of these machines. This means that most students will have to learn about computers via a minimum of hands-on experiences.

The applications in this book are designed to lay the foundations for, and contribute to, computer literacy in the elementary and middle school. But they are not designed to be a complete plan for computer literacy. Computer literacy instruction must include substantial hands-on experience. Students must interact with computers in a variety of circumstances and in a manner designed to promote increased knowledge about computers.

In particular, computer literacy includes learning to use a computer as an exploratory tool to do with as one sees fit. But the teaching of computer programming and the general subject of students being in control of computers is a large and difficult topic—certainly worthy of an entire book, and more. This book does not provide detailed instruction on teaching students to use computers.

Some applications from this chapter are given below. You should note that many are followed by a discussion of the underlying purpose—what students will learn if you use the application in your classroom. If you do not have direct access to a classroom, you should still read each application. This will help solidify your knowledge of the material being covered in the text.



1. Provide the class with a collection of old magazines containing lots of pictures, especially ads. Each student is to prepare a short scrapbook containing pictures of different types of machines. Devote an entire page to each type of machine. Cars, trucks, busses and trains, for example, would all appear on one page. For each page in the scrapbook, the student is to write a brief report on what these machines do, their importance in our society and how they extend the physical and/or mental capabilities of people.

   Machines are commonplace in the lives of all of us, so we tend to accept them without thought. This is a machine-awareness activity, designed to encourage students to think about machines. Since a computer is merely a machine, albeit a very sophisticated one, this is also a computer-readiness activity. Chances are your students will find pictures of calculators and computers, since they are widely advertised. You can gain insight into what your students know about calculators and computers by reading their brief reports on these machines.

2. What can be measured or counted, and what instruments are used? This can be a class project, involving the creation of scrapbook pages or wall posters. Each contains a discussion of something that is measured or counted, along with the instruments that may be used in the process. For example, a thermometer measures temperature. It may consist of a glass tube filled with colored alcohol or mercury that expands when warmed. Temperature is stated in degrees Celsius, where water freezes at 0 degrees C and boils at 100 degrees C.

   A computer is a data processing machine. Often the "data" consists of counts or measurements. Counting and measuring machines may be directly hooked to a computer in order to provide immediate input of the data. Thus, this activity gives insight into computer input devices. Counting, measuring and classifying are fundamental to science, and therefore this activity can be used as part of a science unit. But counting and measuring also occur in sports, in the social sciences, in music, and so on. You can make a substantial contribution to your students' mathematical progress and awareness by emphasizing counting and measuring as they occur naturally throughout the day's curriculum.

3. What are some aids to the human brain? This is a suitable question for small group or class discussion. One category of answers is books, while another is pencil and paper. Other answers are machines such as calculators and computers. The purpose of discussing this topic is to get students thinking about what their brains can do and how it is possible to aid the brain. Is a microscope or a telescope an aid to the human brain? How about eye glasses or a hearing aid? Do students think it is acceptable to make use of these aids?

Exercises

In this book, the Exercises contain activities for the reader. If this book is being used in a formal, college-level course, then some of the exercises can be assigned as homework. If you are studying this book on your own, read through the exercises. They will help to review the material, suggest things you are capable of doing and encourage you to apply and extend the knowledge you are gaining.

1. Select a school district. (If you are an inservice teacher, select your school district.) Determine its policy on teaching about calculators and computers and in allowing their use in the elementary or middle school. Write a brief report on your findings.

2. Select an elementary or middle school. Determine what calculator and computer facilities are available for use by students, teachers, administrators and support staff such as media specialists. How and to what extent are these facilities used? Write a brief report on your findings.

3. Think about how school curriculum is decided. How does a new topic come into the school curriculum and how does an old topic disappear? Write a brief report on the key ideas relating to change in the curriculum.

4. Develop a short questionnaire designed to find out:
   a. What calculator and computer facilities are available in students' homes.
   b. To what extent do students' parents make use of calculators and computers, either at home or on the job.

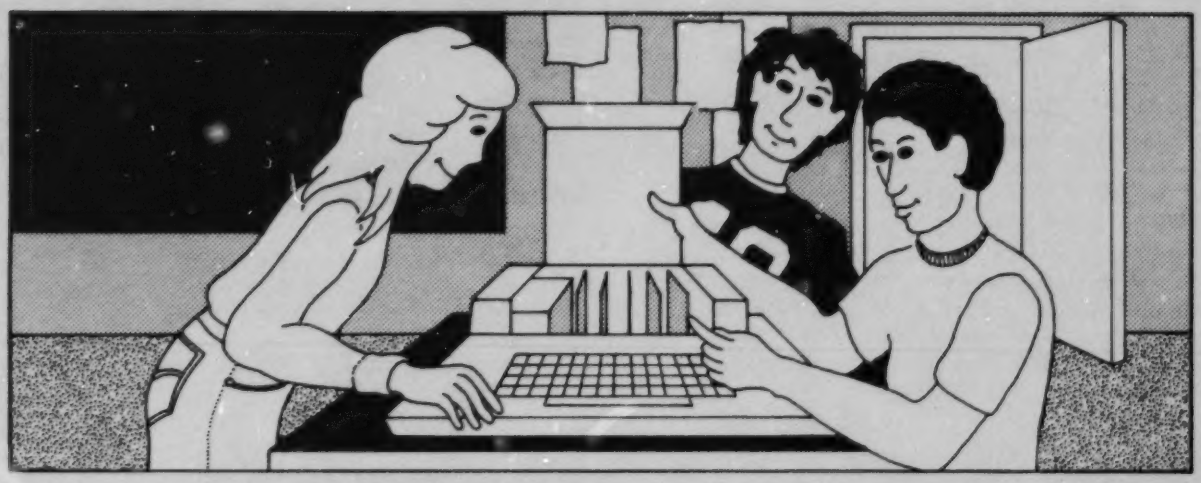
   Use this questionnaire with a class. The students will need to get help from home in filling it out. Discuss the results with the students and prepare a written report on what you learn.

5. Most teacher training institutions require a substantial mathematics course for preservice elementary school teachers. Examine the course you took or the course currently offered by the institution you are attending. Does this course make use of and teach about calculators and computers? Discuss your findings in light of the often quoted maxim that people teach in the way they were taught.

6. Some people feel that calculators and computers will make a substantial contribution to education. Others fear that these machines will dehumanize education and lead to students receiving a poorer quality education. What are your feelings? Explore and express them in light of your current knowledge about calculators and computers. Write down your feelings—but just for yourself. Do not turn in your paper. Rather, save it to return to after you have finished reading this book. You may find that your feelings change with increased knowledge.

CHAPTER TWO

TECHNICAL OVERVIEW



One can learn to use a car, electric light, record player, tape deck, radio, television or telephone without understanding the underlying technology. This is also true with computers. Even preschool children can learn to turn on a computer, load a program into primary storage and run the program. They have no trouble learning to interact with a program written for children their age, designed to hold their interest.

Why, then, do you need a technical overview? Certainly you can learn to use a computer without understanding how it works. Still, there are good reasons for you to know more.

1. Computer technology is changing very rapidly. A good analogy is with the early days of automobiles. In the early 1900s, a car owner needed to be able to service the vehicle and to administer to its particular needs.

   Likewise, the computer field is still quite young, and the development of very inexpensive computers—the ones most apt to be used in an elementary or middle school—is in its infancy. The teacher who ventures to use a computer is still somewhat of a pioneer. Without some technical background s/he can easily flounder.

2. A computer is a quantitatively different type of machine than others that can be easily mastered. The computer has great versatility; it is a general purpose aid to problem solving. To appreciate or understand this versatility, and to make effective use of the computer in problem solving, takes much more than a superficial knowledge.

   Using a computer without understanding the underlying technology is like reading without knowing how to write—it is only half the picture. One is completely dependent upon others to provide material to read. Moreover, it is difficult to appreciate how hard it is to write well without the experience of writing.

3. The magic must be dispelled. Young children often believe that there are small people inside a television set, or they may have trouble understanding how one can send "flowers by wire" through such a thin wire! What computers can do seems like magic even to adults, and some knowledge of the underlying technology is needed to understand the capabilities, applications and limitations of computers.

4. Computers have the potential to cause a major change in both the content and process of education. For this reason alone, all teachers should have a more than casual insight into these machines.

5. Computers are an increasingly important part of our technological society. As an educator you are expected to be an educated person. It is expected that you understand and can cope with varying aspects of life in our society. Thus you should know about technology, and in particular you should know about computers.

The above arguments notwithstanding, at this stage of the history of computers we do not know what

teachers *should* learn about computers. A number of computer-education experts have expressed their opinions via learned papers, books and speeches. These opinions vary considerably, from a need for no knowledge of underlying technology to a need for a sophisticated background in electrical engineering and computer science.

In this book we take a middle-of-the-road approach. This Technical Overview chapter should be fairly easy reading for almost any adult. No knowledge of electronic technology or binary arithmetic is either assumed or taught. The ideas presented here could all be included in a precollege curriculum. Many are suited to the elementary or middle school.

Hands-on Experience

At the end of the first chapter we asked you to obtain some hands-on experience with computers. It is very important that you do this! For some reason, many adults have an unreasonable fear of new technology. Children, of course, do not have this problem.

By now it is assumed that you know how to do the following:

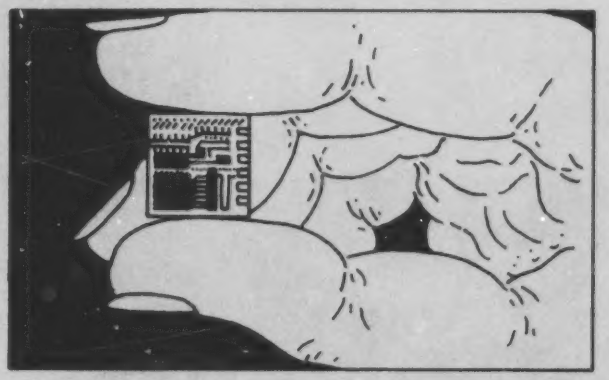
1. Turn on a microcomputer or gain access to a time-shared computer system.
2. Select a program from a program library and load it into primary storage (computer memory).
3. Run the program and interact with it.
4. Turn the microcomputer or timeshared terminal off.

It is also assumed that you have run a half dozen different programs. Thus, you have some insight into both the physical machinery and the computer programs that constitute a computer system.

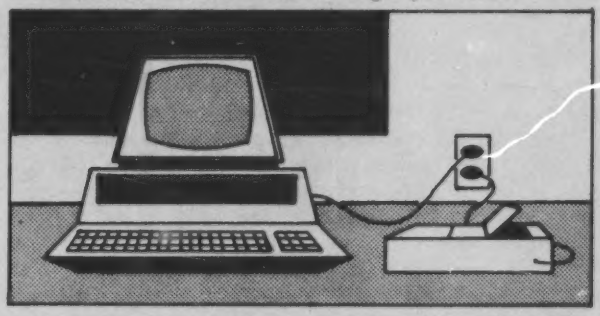
This "how to use it" knowledge comes naturally to children who are in a computer-rich environment. Children who have easy access to computers learn to use them by observing and working with other children, and by observing adults. This is similar to the way children learn to use a television set or a stereo.

Microcomputers

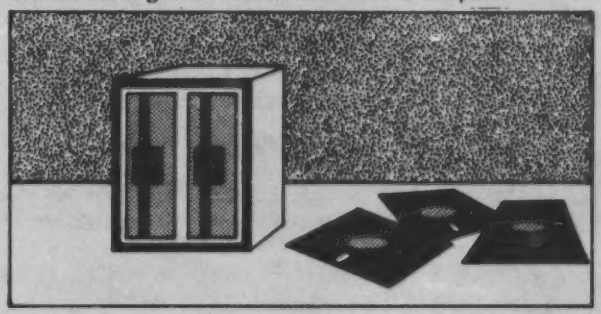
The most widely available computer system in homes and precollege education is called a microcomputer. This is not because of its small size, although a microcomputer may be as small as a portable electric typewriter or a hand-held calculator. Rather, the name comes from the microprocessor, the superspeed automated computational unit that is the heart of the machine. A microprocessor is less than a cubic centimeter in size and may cost less than $10. This little device contains the equivalent of many thousands of transistors, resistors, capacitors and other electronic components. It is called a large scale integrated circuit, or a "chip." The latter name comes from the small piece (chip) of silicon used in the manufacturing process.



The typical microcomputer uses a typewriter-style keyboard for input, a television screen for display of input and output, and some type of magnetic medium for permanent storage of programs and data. The cheapest magnetic medium for storage of programs and data is a cassette tape recorder, which may cost between $30 and $100. Good quality cassette tapes, which will hold several programs or a substantial amount of data, cost less than a dollar if ordered in large quantities.



An alternative to the cassette tape recorder is the floppy disk. A floppy disk is a thin, circular, flexible piece of plastic coated with the same material used on magnetic tapes. There are two standard sizes, a 5¼-inch diameter disk and an 8-inch diameter disk. A floppy disk drive costs between $400 and $1,000. The individual floppy disks cost about $4 each, and hold a substantial number of programs and/or a large amount of data. Magnetic tape and magnetic disk storage will be discussed in greater detail later in this chapter.



The least expensive factory-built microcomputer systems cost about $200. The most expensive large scale computer system may cost $12 million or more. This price range, a factor of about 60,000 between the least expensive and the top of the line, is mind boggling. What other consumer item has such a variance in price? Obviously, there is considerable difference between a $200 computer system and a $12 million computer system. Some of these differences emerge as we study computer hardware in more detail.

Computer Software

A computer system consists of both physical machinery, called *hardware*, and computer programs, called *software*. Both hardware and software are required if the computer system is to function. In this section we give a brief introduction to software by discussing the idea of a procedure which can be carried out by a machine. Chapter 6 of this book is devoted mainly to this very important topic. After this section, the remainder of the current chapter focuses on hardware.

A computer program is a detailed set of instructions which tells a computer what to do. Computer scientists have defined a somewhat more general idea, called a procedure, as follows:

> A procedure is a finite set of instructions (steps) that can be mechanically interpreted and carried out by a specified agent.

A procedure is a specification of how to solve a particular type of problem or accomplish a particular task. Notice that there are three main parts to the definition:

1. *A finite set of instructions or steps;* that is, a detailed step by step set of instructions that can be written into a book or stored in a computer memory.
2. *The instructions can be mechanically interpreted and carried out.* This means the instructions are simple, straightforward, unambiguous. They can be "figured out" and carried out in a machine-like, non-thinking fashion.
3. *An agent is specified.* Computer scientists are especially interested in procedures in which the agent is a computer, or a person working with a computer.

For educational purposes it is instructive to think of the word procedure in a much broader context than just with computers. We learn procedures for tying a shoe, tuning a TV set, cooking an egg, starting a car, looking up a word in a dictionary. Much of education, both formal and informal, consists of learning procedures to cope with specific problem situations. Often the agent in these procedures is a human being, trained to function in a machine-like fashion.

Many of the procedures we learn involve use of machines. You probably know procedures for using a washing machine and clothes dryer to clean your laundry; using a telephone book and a telephone to talk to someone; using a record player or tape player to produce beautiful music; using a calculator to find the quotient of two numbers.

It is interesting to examine paper and pencil arithmetic as a procedure. The agent is a person working with paper and pencil. The person memorizes number facts (the one digit addition and multiplication tables, and so on) and the algorithms. Then the person, with little or no thinking, acts mechanically as part of the agent in carrying out a complicated calculation such as a long division. Continual drill and practice is required to develop speed and accuracy; we strive to make the person increasingly machine-like in this task.

The term "machine-like" perhaps needs some additional explanation. Instruction in arithmetic for very young students includes considerable work with manipulatives (concrete objects). Later, students learn to work with the abstract symbols such as the digits and the operation symbols. Eventually students reach the level where they begin to learn algorithms for working with multidigit quantities. Frequently, a considerable effort is made to motivate and explain these algorithms. So, at least initially, the student may have some insight into how the steps of the algorithm relate to the problem being solved. However, in most cases this is eventually lost. Through continual drill and practice, the student is expected to develop speed and accuracy in tasks such as multiplication, or adding a column of multidigit numbers. The student is expected to become more and more machine-like in carrying out these tasks.

An alternative, of course, is to provide the person with a machine that can do more of the task. A pocket calculator is such a machine. A procedure to find a quotient of two numbers is as follows:

1. Turn on calculator.
2. Key in first number.

3. Push the ÷ key.
4. Key in second number.
5. Push the = key.

The machine then automatically carries out the detailed steps of a long division algorithm and displays the answer.

If you have studied Appendix A of this book or some equivalent material on calculators, you realize the role that computation plays in problem solving. Computation is important, but is no substitute for the human understanding, analyzing, planning, interpreting and applying that goes on in the overall process of solving a problem.

A procedure to be followed by a computer is called a *computer program*. The terms computer software, software, computer program, program, computer procedure and procedure tend to be used interchangeably by computer scientists. There are some problems whose solutions may require following procedures taking millions, or even billions of steps. A medium scale computer system may be able to execute these steps at the rate of a million per second.

It is important to understand that *people* create computer software. Without this software a computer does nothing—solves no problem. But the person who uses a computer need not be the person who created the software, any more than it need be the person who built the hardware. The next time you use a computer, think about the computer program you are using. Keep in mind that it is merely a step by step set of instructions, written by a human, that is being mechanically interpreted and executed by some hardware. A computer system's capabilities are determined by the nature and extent of its software as well as its hardware.

Applications

1. Students at all grade levels know procedures to solve various everyday problems. For example, students get up in the morning, get dressed, eat breakfast and go to school. All of these activities tend to become routine, and so are accomplished with little thought. Have students in the class, individually or collectively, identify familiar procedures. For each procedure, indicate the problem solved and the agent. The agent for waking up in the morning may be a parent, an alarm clock or a neighbor's rooster.

   The idea of a procedure may be new to you, and is likely new to your students. It is a powerful, useful idea that is applicable in all academic disciplines. The above activity, and the ones which follow, are "procedure-awareness" activities. They are designed to help you introduce an important concept into the curriculum.

2. Have your students practice looking up words in a dictionary. Then divide the class into teams of two and give each team a word to look up. One member of the team acts as the agent while the second gives directions. The person giving directions is not allowed to see the dictionary and the person using the dictionary is only allowed to follow the detailed instructions being provided by his/her partner. After a word has been found, switch roles. The purpose of this activity is to help students understand the details of the step by step process we casually call "using a dictionary."

3. Music can be in a written form, a player piano roll, on magnetic tape or on a record. The agent in performing or playing the music is different in each case. Have students identify the agent and discuss the relative quality of the music, with the advantages and disadvantages of each. Then, have students think of other problems which can be solved by a variety of human and mechanical methods. For each, identify the agent. Weaving or knitting provides an interesting example. The purpose of this activity is to help students see that a problem may be solved by a variety of procedures and agents. There are advantages and disadvantages to each procedure.

4. Many students think it is "cheating" to use a calculator. Explore this idea with your students. Have those who think it is cheating explain why. Have students who think it is okay to use calculators explain why. An activity like this helps students to understand other's viewpoints, and to formulate their own.

5. Select an arithmetic calculation at a level appropriate to your students. For example, fifth graders may be doing multidigit multiplication. Give a problem to a pair of students. One student uses pencil and paper, doing exactly what the second student says. The second student gives the directions, telling the first student what to write and where to write it.

   Remember, instructions from the second student to the first must be very explicit. For example, the second student may say, "Write down the multiplication problem 635 times 87," expecting the following:

   ```
     635
   x  87
   ```

   But the agent might instead write:

   THE MULTIPLICATION PROBLEM SIX HUNDRED THIRTY-FIVE TIMES EIGHTY-SEVEN

   Let each student practice in each role. This activity illustrates how complex the arithmetic algorithms are. It should give you some insight into why students often make errors in paper and pencil calculation.

Exercises

1. Suppose you have a set of cards, each containing the name of a student. You know a procedure for alphabetizing this set of cards. Write it down in a form that can be understood and followed by a sixth grader. (This is a difficult exercise; most people have had little practice in writing down the details of a complex procedure.)

2. Repeat Exercise 1, assuming that each student has a different student number, the goal being to arrange them in increasing order. Discuss in what ways your two procedures are alike, and how they differ.

3. A recipe is a procedure for preparing food, while a knitting pattern is a procedure for turning yarn into a garment. Name other types of procedures. For each, specify the usual agent and the problem being solved.

4. The Federal Government provides a detailed set of instructions—a procedure—for preparing a federal income tax return. Learning to be the agent who reads, interprets and executes this procedure is a difficult task. An alternative solution to preparing your own task return is to hire someone to do it for you.

   a. Discuss the educational ramifications of these two methods of solving the income tax return problem.
   b. Give another example with similar characteristics.

   This exercise illustrates a very key idea. There is a difference between understanding the problem to be solved and being the agent that solves it. There are many problems that can be solved by a computer. Thus, the educational issue becomes one of deciding what students should learn to do unaided by machine and what they should be able to do aided by an appropriate machine. This issue has existed since the beginning of formal education, and each new machine adds new dialogue and controversy to it. The computer is such a powerful and versatile machine that it greatly exacerbates the issue.

5. Grayson Wheatley, a well-known mathematics educator, discusses calculators and elementary school arithmetic in an article that appeared in the December 1980 issue of *The Arithmetic Teacher*. In this article, Wheatley indicates that approximately two years of a typical student's mathematics education time in grades 1-9 is spent developing paper and pencil long division skill. Grayson Wheatley recommends that we drop the teaching of paper and pencil multidigit long division from the curriculum and instead give students calculators.

   Discuss the merits of cutting in half the time spent on paper and pencil long division and giving students calculators. The time saved would be used on other mathematics topics such as problem solving.

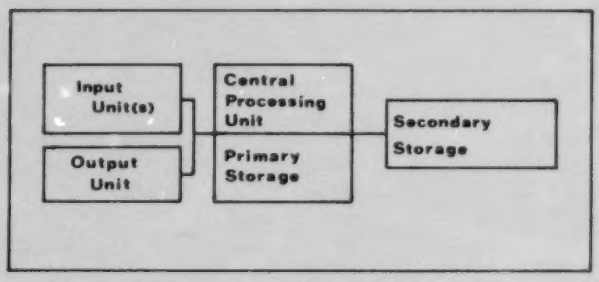
Computer Hardware

Computer hardware is the physical machinery of a computer system. It is designed for the input, storage, manipulation and output of a set of symbols. The set of symbols a particular computer uses is called its character set. The character set for an inexpensive microcomputer may consist of the upper case letters, digits and punctuation marks found on an ordinary typewriter.

A computer's character set may also include lower case letters and/or special graphics characters. The latter may be designed to aid in the construction and output of engineering drawings or pictures.

There are two very important aspects to a computer's manipulative abilities. First, it functions automatically, following instructions from a program stored in its memory. Second, it is fast. A medium-priced computer can execute a million program steps per second.

A diagram of the main hardware components of a computer system is given below. We will discuss each of these five main components. Keep in mind that a computer system may range in price from about $200 to $12 million or more. The more expensive system tends to have more and varied input and output units, greater speed and reliability, and greater primary and secondary storage capacity.



Input Devices

Many computers use an electric typewriter keyboard for input. Anything that can be typed can be input into the computer. The hardware converts the physical motion of a key being stroked into an electrical code for the key's symbol, and stores it in computer memory.

There are many other computer input devices. You are probably familiar with punched cards, often called "IBM" cards. The patterns of holes in a card are codes for characters. A card reader senses these patterns of holes, converts them into electrical impulses and stores the characters in a computer memory.

Optical scanners constitute another important category of computer input devices. The simplest of these read mark sense cards and mark sense scan sheets. You are familiar with the latter, since they are often used as

answer sheets for tests to be machine scored. A more sophisticated example is provided by the bar code on grocery store items. A pattern of bars of varying width and spacing is a code for a manufacturing firm's number and an item number. A laser light is used to "read" the bar code. The reflected light goes into a reading mechanism that translates it into an electrical representation of the characters in the bar code.

Character recognition is now a well-developed technology. Computer input devices that can read a typewritten or typeset page are in widespread use. An entire page can be read in a few seconds. Some of these machines can read carefully handprinted numbers and letters. However, computer recognition of ordinary handwriting is still a research problem. This is not surprising, since many people have trouble even reading their own handwriting!

Another form of character recognition is used on all bank checks processed in the United States. Look in the lower left corner of a check. You will find an account number printed in magnetic ink. This ink contains tiny iron oxide particles. The reading head of a magnetic ink character recognition machine magnetizes these particles and uses a magnetic field detector to read the characters. When a check is being processed by a bank, the amount is printed in magnetic ink characters in the lower right corner by a data entry clerk. Once this has been done, the bank's magnetic ink character recognition machine can read both the account number and amount of the check, so that further processing can be highly automated.

The final input device we will consider is the general category of sensing devices. Sensing the movements of a knob or lever in a hand control for a TV video game provide a common example. Devices can be built to sense temperature, pressure, velocity or acceleration. These measurements are converted into digital form and input to a computer. The computer can then be used to control a process. For example, a computerized thermostat system senses the temperature at a number of places throughout an office building. The computer, taking into consideration the time of day, weather conditions and number of employees, opens and closes air vents and adjusts air condtioners and furnaces.

There are many more input devices, some especially useful in education. Examples include voice input, touch sensitive display screens, light pens, graphics tablets and television cameras. Each has educational value and educational implications. For example, consider a touch sensitive screen and/or voice input. Each can be used with non-readers. They may be quite useful in teaching young students to read or in working with handicapped students.

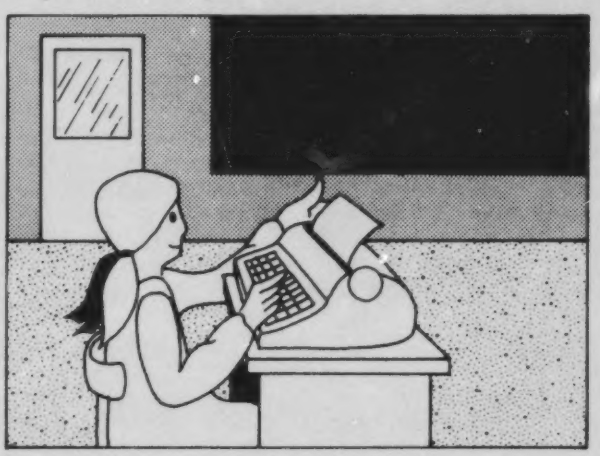
At one time the cost of touch panels, voice input, graphics tablets, etc. was quite high. But that time is now past. More and more we will find that these are available to computer users throughout business and education, and at home. As appropriate educational software is developed to take advantage of these computer hardware facilities, we can expect their use to grow rapidly.

Applications

1. Most grocery store items are labeled with a bar code, designed to be read by a laser device. Usually the numerical values represented by the bar code are printed below the bars. Each major food processing company in the United States has a code number, and each food item it produces has a code number. Thus, when this information is input to a computer, it can "look up" the current price and ring it up on the cash register.

   Have students collect labels containing bar codes from a large variety of food items. Help them to discover that each company has a unique number, and that different food items from a company have different numbers.

   Discuss with the students how the bar code helps the grocery checkout clerk do his/her job, and other possible applications such as inventory control and daily sales analysis. Discuss disadvantages of having such a checkout system in a store. For example, a clerk using this machinery may require less training and experience than one using less sophisticated equipment, and consequently receive a lower rate of pay. A computer awareness activity such as this can easily lead to serious discussion of the social consequences of computers.



APPENDIX A

CALCULATORS

Calculators are of increasing importance in schools, homes, and places of business. It is likely that you own one and have had considerable experience in using it. The inexpensive electronic pocket calculator has many features in common with a general purpose computer. Indeed, one way to begin one's study of computers is via a course on calculators. This Appendix provides a brief overview of some of the key ideas one might find in a calculator course for elementary and middle school teachers. If you have not studied the material of such a course, then an hour or so spent on this Appendix will help you considerably as you begin to learn about computers.

Most inexpensive calculators use algebraic logic, and it is very easy to learn how to use such a machine. To divide 2497.608 by 83.7, one merely keys

[2] [4] [9] [7] [.] [6] [0] [8] [÷] [8] [3] [.] [7] [=] .

The typical adult learns to add, subtract, multiply, and divide on such a machine in a minute or two. (Grade school students may take a little longer to master calculators to this level, but even for them learning to use a calculator is "no big deal.") We assume that you have this level of calculator knowledge. Also, we assume you understand the use of the CLEAR key [C] and the CLEAR ENTRY key [CE] .

However, an elementary or middle school teacher needs to know more about calculators if s/he is to make effective use of them for instructional purposes. A detailed treatment of this topic is given in the book *Calculators in the Classroom: With Applications for the Elementary and Middle School Teacher*, by David Moursund (published by John Wiley and Sons, 1981). The ideas which follow are all covered in that book.

Problem Solving

The most important goal of mathematics education is to help students learn to solve math problems. A calculator is designed as an aid in solving certain types of math problems. Many of these problems can be solved by following a five-step plan. Examine the steps given below, and decide for yourself where you think a calculator is a useful aid.

1. Understand the problem.
2. Represent the quantities, knowns and unknowns, relevant to the problem.
3. Plan a method of attack.
4. Execute (that is, carry out) the plan.
5. Examine the results for correctness and meaningfulness.

Calculation may be necessary or useful at every step. However, the greatest amount of calculation is apt to occur in the execute step (step 4), and it is here that a calculator is most useful.

It is important to realize that computation by itself solves no mathematics problem. A calculator is in no sense a replacement for a mathematics curriculum. But a person equipped with a calculator has a considerable problem-solving advantage over others who do not use a calculator.

Modes of Calculation

Commonly used modes or methods of calculation include:

1. Mental arithmetic, exact or approximate.
2. Pencil and paper arithmetic.

3. Use of math tables, including tax tables, navigational tables, sales tax tables, and so on.
4. Use of machine aids such as the abacus, slide rule, calculator and computer.

The role of pencil and paper needs to be made clear. First, all pencil and paper algorithms are actually mental arithmetic algorithms, in which one uses pencil and paper to record intermediate results and final results. Second, pencil and paper are indispensible aids to mathematical problem solving even if one has unlimited use of a calculator. Imagine trying to develop a plan of attack on a complicated problem without being allowed to draw diagrams, write down your ideas, write down formulas, and so on.

In any event, mental arithmetic is by far the most commonly used mode of calculation. It has been estimated that 75% of all arithmetic an ordinary person needs to do in everyday life can easily be done mentally. Most teachers agree that our current mathematics curriculum is especially weak in helping students to develop good mental estimation skills. All students should develop good mental arithmetic skills, both exact and approximate. This is true regardless of how readily available calculators may become. Mental calculations are quicker and more convenient than use of a calculator in a majority of computational situations.

Changes in technology change the computational aids people use. Both the abacus and the slide rule are of declining importance, while the calculator and computer are of increasing importance. Such changes can have a significant impact upon the curriculum. Mathematics education leaders in the United States are quite supportive of the use of calculators in education. The National Council of Teachers of Mathematics has been supportive of calculator usage since the mid 1970s. In 1980 they issued a statement on math goals for the 1980s in which they placed very strong emphasis on the use of calculators and computers.

Mathematics education leaders are suggesting decreased emphasis upon pencil and paper computation. The time saved could be spent gaining increased mental arithmetic skills, learning to use machines such as calculators and computers, and developing better problem-solving skills.

Getting Correct Answers

We frequently read about computer errors, where a computer is blamed for sending out incorrect tax bills or for printing incorrect payroll checks. But we never read about calculator errors in the same vein.

A calculator, like a computer, is a machine. The machine can malfunction; its circuitry can calculate incorrectly. But most calculator errors are actually human errors—the fault lies with the person using the machine. (It turns out that the same holds true for computer errors.)

Each time a person uses a calculator s/he should test the machine. This is done by performing some calculations mentally and on the machine, and comparing results. Select calculations to test the [+], [−], [×], [÷], [C], [CE], and digit key circuitry. If the machine checks out okay on simple calculations, then it is highly probable that the circuitry will perform correctly on all calculations.

It is easy to err when using a calculator. One reads numbers or operations incorrectly, pushes wrong keys, makes an order of operation error, etc. The order of operation error is frequent in chain calculations such as [3] [+] [2] [÷] [5] [=]. Some calculators will give 1. for the answer to this calculation, while others will give 3.4. That is, some will calculate (3 + 2)/5 while others calculate 3 + (2/5). The calculator user must understand what calculation is to be performed and how the particular calculator being used actually functions in a chain calculation.

Most calculator errors can be detected by one of the following methods:

1. Do it twice. Do each calculation twice, working in a different order if possible. (Check division by multiplication.)
2. Use mental arithmetic.

The latter is most important. If one understands the problem being solved, thinks about the meaning of the numbers, and makes appropriate mental approximations, then most calculator errors can be detected.

The key point is that a calculator is a machine. A person who chooses to use a calculator should be aware of its capabilities and limitations. It is the person, not the machine, who must take ultimate responsibility for wrong answers. The person must understand the problem being solved well enough to detect errors, whether human or machine produced.

Calculator Functions

A four-function calculator performs the four functions commonly called addition, subtraction, multiplication, and division. Many calculators also perform the functions $\sqrt{\ }$ (square root) and % (percent). It is possible to build a handheld calculator that can perform 50 to 100 or more different functions at the press of a key or two. Some of these, such as squaring a number or taking the reciprocal of a number, are suitable for study by elementary and middle school students. Others, such as trigonometric and statistical functions, are not of interest at this level.

A function is a mapping from one mathematical set into another. One can speak of input to a function and output from a function. A calculator function takes one or more numbers as input and produces a number as output. Thus, addition is a function with two inputs, while square root requires a single input.

The idea of a function is one of the most important ideas in all of mathematics. Foundations for learning about functions are laid in elementary school mathe-

matics instruction. A calculator, and instruction about calculators, can be a valuable aid in helping students learn about functions.

Closely related to the concept of a function is the concept of a formula. A formula such as A = LW or P = 2L + 2W represents a plan for solving a particular problem under consideration. A formula can be used to determine an unknown value from known values. For example, the area and perimeter of a rectangle can be determined if the length and width are known. For a formula such as P = 2L + 2W, one can think of inputting the values of L and W, and producing as output the value of P. This is similar to what a function does.

Functions and formulas are key aspects of computer programming; work with functions and formulas provides a foundation quite useful in eventually learning to program a computer. But alternatively, learning to program a computer provides a foundation for the study of functions and formulas. As instruction in computer programming becomes more common in elementary and middle schools, there will need to be substantial changes in mathematics education at higher grade levels.

Calculator Memory

Every calculator has two kinds of memory—temporary and permanent. Consider the keying sequence [4] [+] [3] [=] . The numbers 4 and 3, and the operation +, are stored temporarily in memory. Pushing = tells the machine to carry out the (stored) operation on the (stored) numbers. It does this by following a detailed step by step set of directions stored in permanent memory. Algorithms for +, −, ×, ÷, and all the other built-in functions are stored in permanent memory. Many calculators have additional temporary memory and keys such as [M+] , [M−] , [CM] , and [RM] . Such extra memory is useful in dealing with fractions and in multi-part calculations such as

$$\underbrace{89.74 \times 63.8}_{\text{Part 1}} + \underbrace{3897.5 \div 43.2}_{\text{Part 2}}$$

The calculation for Part 1 is completed and the result is stored in a memory location. Then the Part 2 calculation is performed. The result is combined with the previously stored answer to complete the calculation.

A calculator may also have permanent or temporary external memory. This may be a plug-in cartridge, a magnetic card, or a magnetic strip. It is possible to have a library of programs (detailed step-by-step sets of directions) that can be input to the calculator as needed. A calculator with this feature is called a programmable calculator. (Note that some inexpensive programmable calculators do not have provisions for an external memory. Programs may be stored in written form and keyed into the calculator when needed.) All of these calculator memory ideas carry over to computers.

Calculator Arithmetic

The calculator number line is limited in length and has only a finite number of points. This is quite different from the real number line. Thus, as is to be expected, calculator arithmetic is not always the same as real arithmetic. Each of the aspects of calculator arithmetic discussed below carries over to computer arithmetic.

As a simple example, 10 ÷ 3 = 3.3333333 on an 8-digit calculator. While this difference from real arithmetic is small, it can be significant in some problems. One place such differences may be encountered is in checking division problems. On a calculator 47.6 ÷ 23.7 = 2.0084388. When the answer is multiplied by 23.7 the result is 47.599999. One must recognize that this is nearly 47.6 to see that the calculator-produced division answer is probably correct.

The finite length of the calculator number line allows a calculation to overflow, producing an answer that lies outside this line. Also, it is possible that a calculation will unexpectedly produce an answer of 0. For example, on an 8-digit machine not using scientific notation, the calculation .0002251 × .0000084 and the calculation .00037 ÷ 89438 each produce 0. This is called an underflow.

Scientific Notation

Scientists often encounter very large numbers and very small numbers. They write these as a decimal fraction part times a power of 10.

$$84{,}692{,}000{,}000{,}000 = 8.4692 \times 10^{13}$$
$$.000000003594 = 3.594 \times 10^{-9}$$

Some calculators are designed to work with scientific notation numbers. Often such calculators are called "slide rule" calculators and contain a number of built-in trigonometric functions and other functions from higher mathematics. The number line on such a calculator may extend from -9.99999×10^{99} to $+9.99999 \times 10^{99}$. While this line is longer than the number line on a simple 8-digit calculator, the problems of overflow and underflow can still occur.

A computer may make use of several different number systems. A scientific notation number system is an inherent part of Basic and many other programming languages.

Guess and Check

The methods of problem solving we teach in schools are dependent upon the modes of calculation available. One general-purpose method of problem solving, called guess and check, may involve a large amount of calculation. Thus, it is highly desirable to have a calculator or computer available when using the guess and check method. To illustrate guess and check, suppose you needed to know the cube root of 17 ac-

curate to one decimal place. You could proceed as follows:

Step #	Guess	Calculation Mode	Result	Conclusion
1	2	Mental	$2^3 = 8$	Too low
2	3	Mental	$3^3 = 27$	Too high
3	2.5	Calculator	$2.5^3 = 15.625$	Too low
4	2.7	Calculator	$2.7^3 = 19.683$	Too high
5	2.6	Calculator	$2.6^3 = 17.576$	

We conclude by systematic guess and check that 2.6 is the desired answer.

Using a calculator, one may make several dozen trials to solve a problem. The amount of effort required, or the amount of time it takes, is not too great because a calculator is quite fast.

A computer may be a million times as fast as a simple calculator. Using a computer it is reasonable to make many millions of guesses when using guess and check to solve a problem. The ability to systematically make millions of different trials is a very powerful aid to problem solving. As a consequence of calculators and computers, the guess and check method of problem solving should have increased importance in the curriculum.

Calculators in Education

Some people mistakenly believe that the main goal of mathematics education is to acquire a high level of skill in paper and pencil calculation. To such people the calculator is a considerable threat. What will the students do if they are not spending their daily math time practicing paper and pencil calculations?

The answer is that they will have more time to learn mathematics and mathematical problem solving. They will have more time to develop mental arithmetic skills, both exact and approximate. They will have more time to learn about calculators, calculator functions, and calculator memory. They will have more time to learn about computers.

Pencil and paper arithmetic will not and should not disappear. Indeed, it is to be preferred over a calculator in simple additions and subtractions such as in keeping score for a bridge game. But clearly this mode of calculation is decreasing in importance. As with any change in education, the change process is slow and difficult.

A calculator is a very modest tool as compared to a computer. But even calculators can serve as a basis for a substantial change in mathematics education. For this change to occur, elementary and middle school teachers must become knowledgeable about the capabilities and educational ramifications of calculators.

2. Ask students in your class if they have ever used a typewriter. Have them tell about it, and whether they enjoyed it. Bring a typewriter to your classroom. Let students use it to type out their names and/or a short story. Have students compare their typing speed with their writing speed. Point out that most people can learn to type 50-60 words per minute, given adequate training and practice.

   An activity such as this serves several purposes. It familiarizes students with a common machine, one which they may not have previously encountered. It is a "fun" machine, so most students will enjoy using it. A keyboard terminal is a common computer input device. Thus, this is a good computer-readiness activity. The measurement of writing and typing speeds is a good application of mathematics and of timing devices.

3. Each column on the standard 80 column punch card can contain one character. Digits are coded by a single punched hole, letters by two holes and special characters by three holes. Give each student a decoder card punched with the digits, letters and special characters and printed along the top. Then give each student a card containing a message that has been punched, but not printed on the top of the card. The task is to decode the "secret" message. The same decoder card can be used to decipher bills the students' parents receive on punched cards.

   The main purpose of this activity is to familiarize students with punched cards and the idea that characters can be coded as patterns of holes. The patterns of holes are easily read by a machine, while a person more easily reads the printed characters on the top of a card.

4. Divide students into two-person teams and give each team a calculator and a set of calculations. The team member with the calculator serves as the calculator's ears and voice. This team member can receive and carry out spoken instructions such as:

   TURN ON
   ENTER FIVE THREE POINT SEVEN
   ENTER TIMES
   ENTER FOUR POINT SIX
   ENTER EQUALS
   OUTPUT ANSWER

   One team member reads a problem, gives the instructions, and records the final answer. The other uses the calculator.

   This activity gives students practice in giving and receiving instructions. It illustrates how a voice input and voice output calculator or computer might work.

Exercises

1. A good typist can type 60 words per minute, figuring an average of five characters per word. A punched card can contain 80 characters, and a high speed cardreader can read 1000 cards per minute. How many words per minute is this, and how many times as fast as a good typist is it?

2. Suppose that material has been typed double spaced, so there are about 72 characters per line and 30 lines per page. An optical scanner can read this material at the rate of one page every three seconds. Compare this speed with that of the typist and the cardreader of Exercise 1.

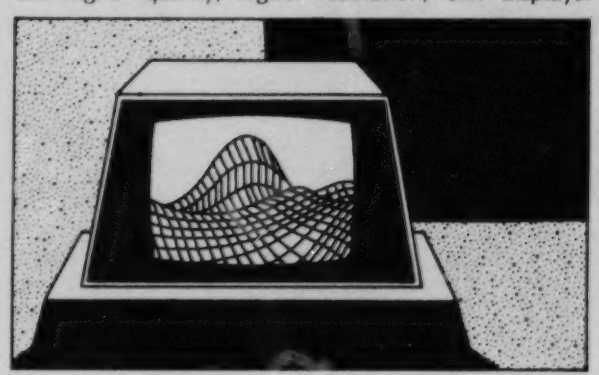
3. A good (human) reader can read 500 words per minute. Compare this speed with the cardreader of Exercise 1 and the optical scanner of Exercise 2.

4. Many grocery stores now have electronic scales connected to their cash registers. Talk to a grocery store clerk to learn how these machines work. Write a brief report on what you learn, including the clerk's attitude towards this computer technology. Does the existence and widespread use of electronic digital scales have a potential impact upon the content of the elementary or middle school curriculum?

Output Devices

An electronic typewriter can be modified to serve both as a computer input device and output device. Such a keyboard terminal prints the computer output on ordinary sheets or rolls of paper, and is called a hardcopy terminal. The term "hardcopy" is used to distinguish this permanent printed output from a non-permanent screen display, sometimes called softcopy output. Hardcopy terminals often use other forms of printing mechanisms and may require special paper, such as heat sensitive paper.

The least expensive screen display keyboard terminals use a television set, or a modified television set, as a display mechanism. The display screen of a TV set is a cathode ray tube (CRT). CRT's are also used in oscilloscopes and in other electronic display mechanisms. Higher quality screen display keyboard terminals use higher quality, higher resolution, CRT displays.



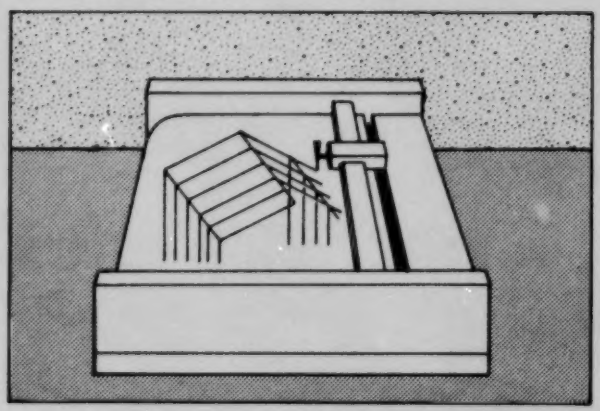
With a higher resolution display, one can draw intricate architectural or engineering drawings, graphs and pic-

tures. The field of computer graphics has emerged as an important aspect of computer science.

The TV set used as a computer output device may be a black and white set, but it may also be a color set. Indeed, color output is now quite common. This certainly adds a new dimension to computers in education. Some of the educational materials now available are more than adequate competition for the Saturday morning cartoons on TV!

The printing speed of a hardcopy keyboard terminal can vary widely, depending upon price and quality. The old fashioned model 33 and model 35 Teletypes, once widely used as computer terminals, print 10 characters per second. Many of the newer hardcopy keyboard terminals have printing speeds of 30 to 60 characters per second, or even higher. A typist's speed is often stated in words per minute, figuring an average of five characters per word. Thus, a printing speed of 10 characters per second is two words per second, or 120 words per minute. A printing speed of 60 characters per second is equivalent to typing 720 words per minute, or about 10 times the speed of a good typist.

But such printing speeds are slow, relative to those available on line printers. A line printer prints an entire line nearly simultaneously. Printing speeds of 300 lines per minute up to 2000 lines per minute are possible on impact printers. In an impact printer, a hammer presses an inked ribbon against the paper, as in a typewriter. Use of a xerox (non-impact) printing process allows speed up to 20,000 lines per minute. Line printers generally have long carriages, allowing lines longer than will fit on a notebook-sized page. A line length of 132 characters is common. Thus, a line printer may print tens of thousands, or even hundreds of thousands of words per minute.



Scientists, engineers and architects frequently make use of a computer output device called a plotter. A plotter contains one or more pen mechanisms. Under computer control, the pen is quickly positioned at desired spots on a paper, raised or lowered, and moved rapidly. (Some plotters move both the pen and the paper, while others move only the pen.) Pen movements as short as a hundredth of an inch, with several hundred pen movements per second, are possible. A plotter may have several pens with different colors of ink. Quite intricate and colorful patterns or drawings can be output by a plotter.

There are many other types of computer output devices. For example, a computer can prepare punched cards, print on microfilm or on larger photographic plates to be used in photo typesetting, run a Braille writer, output music, or even output voice. Also, computers can be used to control devices such as furnaces, air conditioners, all kinds of factory machinery, airplanes and rockets. A large part of all the computer equipment currently being produced is actually being used in such process control applications.

Sophisticated input and output devices can be combined to make tools especially valuable in education of the handicapped. A hand-held, battery powered "talking" calculator has been commercially available since the latter part of the 1970s. As each key is depressed, the calculator speaks (provides voice output) of the key's symbol. After an answer has been calculated, the calculator can be directed to speak its digits. Needless to say, the talking calculator is quite useful to blind students.

As another example of use of sophisticated computer technology, one can now purchase a reading machine, designed to be used by blind people, which has a TV camera as an "eye" for reading books and other printed material. It uses a voice synthesizer to speak what it reads. This machine, developed in the latter part of the 1970s, is an excellent example of how computer technology can make a very important difference in the lives of many people.

Applications

1. Ask your class if they have ever seen a robot on television or in a movie. Get them to list the robot characters they have seen. (Likely they will incorrectly include Steve Austin and Jamie Summers, who are merely "bionic.") Discuss with the class the input and output mechanisms of these robots. Help them to understand that these are fictional characters and that real-world computerized robots do not have such verbal and aural skills or intelligence.

   The purpose of an activity such as this is to help students begin to distinguish fiction from reality. There are many aspects of computers that seem almost like fiction even to computer scientists who work with them on a daily basis. Thus, it is not surprising that fiction is confused with reality by most children and many adults.

2. High speed computer printers are often used to print personalized ads or personalized books. These are form letters or standard short stories in which the person's name is inserted at a number of different places. Ask students to bring examples of these to school. Students may be able to get samples of such

direct mail advertising from their parents.

The purpose of this application is to help students become aware of how impersonal the computer actually is in this type of application. Printing a person's name at various places in a form letter really does very little to personalize it.

3. Computer art and computer architectural or engineering drawings are now common. Have students look for examples in magazines, especially in ads. Computer art contests are now held periodically, so examples may be found in newspaper stories or magazine articles. Some students may have shirts imprinted with pictures of themselves, done by a computer. Help the class to create a bulletin board display of these types of computer output. Use it as a basis for discussing whether computer art is really "art."

4. The idea of measuring performance speeds in words per minute will interest many students. After discussing and illustrating the idea, you may have students determine their printing speed, cursive writing speed, reading out loud speed and silent reading speed.

   Students may also want to determine the words per minute speed of a 20 page per minute copying machine, a 2000 line per minute computer printer, or a 20,000 sheet per hour newspaper press.

Exercises

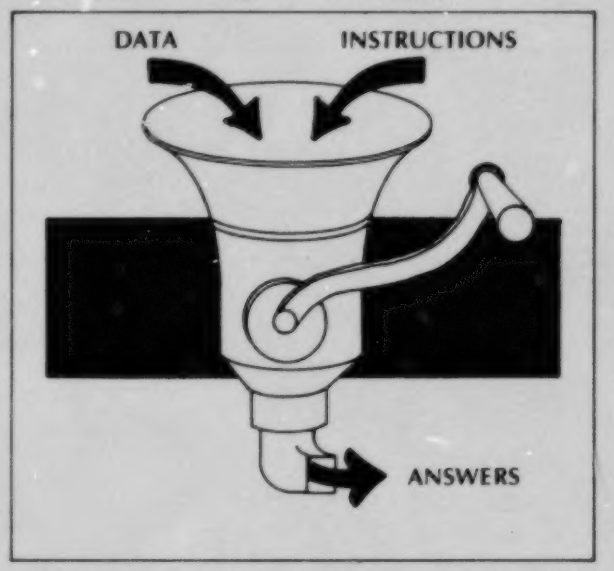
1. Suppose that a line printer has a 132 character line length. How many words per minute, figuring five characters per word, is a line printer speed of 300 lines per minute, 1500 lines per minute, and 20,000 lines per minute? How many times faster than a 60 word per minute typist is each of these?

2. Computerized phototypesetting is now common in the newspaper industry and in many other printing shops. In the past decade there have been many prolonged newspaper strikes by unions opposed to the introduction of this equipment. Study this aspect of computers in automation and write a report on what you learn. (Be aware that the linotype operators, whose jobs were being eliminated, were highly skilled, very experienced, and often well educated.)

Central Processing Unit

The heart of a computer is the central processing unit (CPU). The CPU has two maior components. First, it contains a control unit that can read and understand an instruction (a program step) that is in primary storage. Second, it contains an arithmetic-logic unit, which can be thought of as a superspeed calculator.

The CPU operates in a two-step cycle. In the first step it fetches an instruction from primary storage (computer memory) and interprets its meaning. In the second step it executes the instruction. The execute step may require fetching additional data from primary storage. For example, an instruction may indicate that the numbers in two specified locations are to be added. During the execute cycle the two numbers are brought into the arithmetic-logic unit and added.



The most impressive aspect of a CPU is its speed. The time to carry out a fetch cycle or an execute cycle is measured in microseconds (millionths of a second) or nanoseconds (billionths of a second). A relatively inexpensive microcomputer can carry out a fetch cycle in a microsecond, while a more expensive computer may be 10 to 100 times this fast. The execution of a simple instruction on a microcomputer may take several microseconds, while a more expensive computer may be 10 to 1000 times as fast. The very fastest modern computers can execute an instruction in 1.25 nanoseconds.

It is difficult to comprehend the meaning of the speed of a CPU. A medium scale computer can carry out a million instructions in a second. An instruction might be an arithmetic operation (+, −, ×, ÷), a logic operation (compare two quantities to see if they are equal) or a movement of data between various memory locations. How long do you think it would take you to carry out a million arithmetic operations either mentally or using pencil and paper? A pencil and paper long division may take a person a minute or so. During this time a computer can do more arithmetic than a typical person does in a lifetime.

Other examples add to this insight. You can walk perhaps 5 km/hr. Thus, to walk across the United States would require several months. If your walking speed increased by a factor of a million, you would be able to cross the United States in less than four seconds! Or suppose that your reading speed increased by a factor of a million. You would then be able to read several thick novels in less than a second.

Applications

1. Give your students a set of arithmetic fact calculations such as one-digit addition or multiplication that they can do rapidly. Time them as they do a set of these calculations and help the students express their speed in calculations per minute. Have them determine their number of errors per minute.

   A medium-sized computer can perform 60 million of these calculations per minute with no errors. Have each student compare his/her speed to that of a computer.

   This activity involves the student in measuring speed and accuracy and in comparing his/her work with that of a computer. It leads to some appreciation for the immense speed and exceptional accuracy of a central processing unit. The next exercise has similar goals.

2. A computer may be a million to ten million times as fast at calculation as a person equipped with a calculator. What does it mean to be a million times as fast?

   a. Help students to measure their reading rates, and to estimate how long it would take to read a certain book. Then suppose they could read a million times as fast. Help them to determine how long it would take to read the book, or to read every book in the school library.

   b. Help students to measure their normal walking speed in km/hr. Then help them to estimate how long it would take to walk across the United States (5,000 km) or around the world (45,000km). Then suppose they could walk a million times as fast. Help them to estimate how long it would take to walk across the U.S. or around the world.

   c. Give each student a small jar of beans and time how long it takes to count them. Suppose the entire classroom were full of such beans. Help the students to estimate how long it would take to count a roomful of beans. Then help them to estimate how long it would take if they could count a million times as fast.

3. Exercises 2 and 3 below can be used at the upper elementary or middle school level. Encourage your students to look for other examples which would show the great speed of a computer.

Exercises

1. Carry out Application 2a and 2b for yourself.

2. Select a large dictionary and estimate how many words it contains. Suppose you were going to look up a word by comparing it with each word in the dictionary, starting at the beginning of the alphabet. That is, you would look at every word beginning with A, then every word beginning with B, and so on. Estimate how long it would take you to find a specific word beginning with L. Then estimate how long it would take if you could read and compare words at a million times your current rate.

3. Time yourself as you do a calculation such as 38.9×614.3 using pencil and paper. Estimate how long it would take you to do a million multiplications of this complexity, working eight hours per day, seven days per week.

   Suppose that a computer which can do a million multiplications per second rents for $200 per hour; what would the cost be to use this machine to do a million multiplications? Suppose that you were paid the same amount to do a million multilications using pencil and paper. What would be your hourly rate of pay?

Computer Memory (Primary Storage)

Every computer has a memory—space where programs and data can be stored either temporarily or permanently. For most computers the memory is divided into two categories: primary storage and secondary storage.

Primary storage operates at a speed comparable to that of a CPU. This is necessary because when a program is being executed there is continual interaction between primary storage and the CPU. For example, an instruction may need to be brought into the CPU. An answer calculated by the CPU may need to be placed into primary storage.

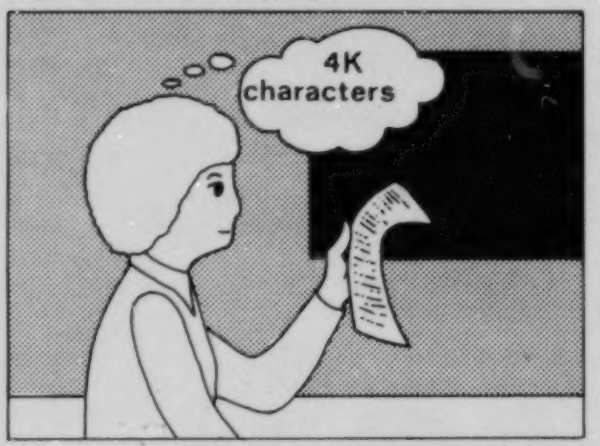
One way to state the speed of a computer is to give its memory cycle time. This is the time to move a character from primary storage into the CPU; it is also the time to move a character from the CPU into primary storage. Typically, the CPU can execute a simple instruction, such as to increase a particular number by one, during this memory cycle time. Even a relatively inexpensive microcomputer may have a memory cycle time of 500 nanoseconds. Such a machine can count from 1 to 2,000 by 1's in a thousandth of a second!

In addition to giving the memory cycle time, a standard way to describe the size or capability of a computer system is to give its memory size. The unit of measure of computer memory is the byte, which is the same as one character of storage. A digit, letter or punctuation mark can be stored in one byte of memory. Computer scientists frequently use the letter K to stand for the number 2^{10}, which is 1024. They state the size of a computer memory as a number of K's of storage.

For example, an inexpensive microcomputer is apt to have an 8K or 16K primary storage. An 8K machine can store $8 \times 1024 = 8192$ characters in its primary storage. A medium scale computer may have a 256K or 512K primary storage, while a large scale computer will have primary storage of 1024K (called a megabyte) or several megabytes. The larger the primary storage, the larger

the program and the amount of data that can be immediately available to the CPU.

One can get a feeling for the size of primary storage by comparing the numbers with the amount of print on a typed or printed page. A full page of single spaced typing is about 4K characters. Thus the primary storage of an 8K machine is roughly equivalent to two pages of single spaced typing. The primary storage of a four megabyte machine is roughly equivalent to a thousand single spaced pages of typing.



Primary storage is temporary storage; what is stored there as a person runs a program is replaced by the program and data of the next person to use the machine. Also, the contents of primary storage are usually erased when the machine is turned off. For these reasons, almost all computer systems have some type of more permanent, large capacity secondary storage. This is discussed in the next section.

Applications

1. When written material is stored in a computer memory, each character, including the blank space between two words, occupies one byte of storage. Students can develop an understanding of the amount of material in a book, magazine article or newspaper article by counting or estimating the number of characters in a variety of examples.

   a. Estimate the length of a book by counting the number of characters on one page and multiplying by the number of pages. Do this three different times, using the second, tenth and 25th pages of the book. Are all of the estimates the same? Explain.

   b. Estimate the number of characters on the front page of a newspaper by counting the characters in a 10 cm column, measuring the number of cm of printing on the page and doing appropriate arithmetic. Compare the contents of the front page of a newspaper with those of one page of the book used in (a.) above.

   c. Estimate the number of characters in a set of encyclopedias. How many times larger is this number than the length of the book used in (1a) above?

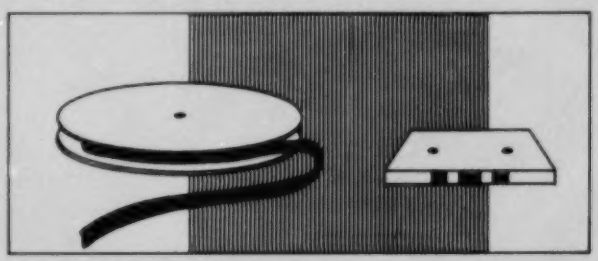
   This exercise, like many others in this book, helps to develop and encourage quantitative thinking. Many things can be counted, measured or in other ways quantified. This is a key aspect of science, and of the computer field. Often, it is also a valuable application of arithmetic.

Exercises

1. Select an adult novel or college level textbook. Use it in Application 1a above. Would the contents of your book fit into the primary storage of a one magabyte computer?
2. What is the binary number system? Why do you think that computer scientists use 2^{10} as a measure of memory size, and that typical memory sizes are 8K, 16K, 32K, 64K, etc.?

Computer Memory (Secondary Stage)

Secondary storage, often called bulk storage, is for more permanent storage of large quantities of data and programs. The most common and widely used forms of secondary storage make use of magnetic tape, magnetic disks or other related magnetic media. The magnetic tape used on computer systems is similar to that used on hi-fi systems, but may be of higher quality.



Inexpensive microcomputer systems frequently use a tape cassette recorder for secondary storage. These recorders, costing perhaps $30-$100, are widely available in radio/television stores. A fifteen minute tape can store about 32K to 64K bytes, depending on the computer being used. However, these cassette systems are not particularly reliable, and loss of programs or data is frequent.

More expensive tape systems, costing hundreds or even many thousands of dollars, are often used on more expensive computer systems. An industry-wide standard type of tape is ½ inch wide and comes in reels of 600, 1200 and 2400 feet in length. Recording densities of 800, 1600 and 6250 bytes per inch are common. A 2400 foot reel of tape, recorded at 6250 bytes per inch, stores 180 million characters.

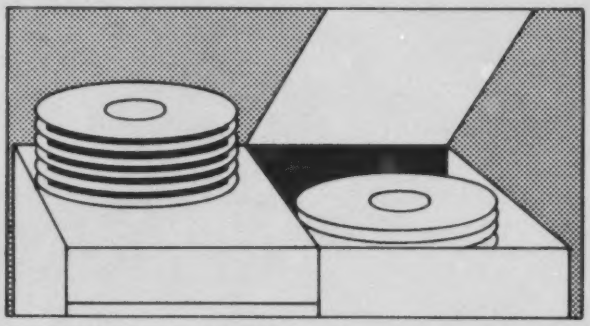
The storage of millions of characters is somewhat

difficult to comprehend. The typical mystery book or short novel is about one-half million characters in length. A 500 page pocket book (a full length novel) is about a million characters.

The 2400 foot reel of tape is about 30 cm in diameter. Recorded at 6250 bytes per inch, it can store the equivalent of about 180 full length (500 page) novels. Such a reel of tape costs about $15. It is a highly reliable, long-life storage medium. With proper care and not too frequent use the tape will last for several years.

Storage on magnetic tape has one serious drawback. A tape must be read sequentially. If the program or data one wants is near the end (center) of a 2400 foot reel it can take many minutes for the reading mechanism to reach it. Suppose, for example, a bank kept its customer account records on a tape. A customer requests information about his/her account. Several minutes of tape drive time may be used to provide the answer.

An alternative to this is magnetic disk storage. Flat circular aluminum plates are coated with iron oxide, the same material used on magnetic tape. The plate, called a disk or a hard disk, is spun rapidly. A read/write arm is quickly positioned over any spot on the disk. Thus, access to any program or data on the disk occurs in well under a second.



A disk pack consists of a number of disks mounted on one spindle with air space between them for read-write arms. In a disk pack the top and bottom of each plate is used as a storage medium, except for the top of the topmost plate and the bottom of the bottom-most plate. An eleven plate pack, with 20 recording surfaces, may store 300 million to 600 million bytes or more. A single recording surface of a hard disk may store anywhere from 15 to over 30 million bytes.

Again, let's try to understand what these numbers mean. The disk pack is about 35 cm in diameter and 15 cm high. It costs about $300. It stores the equivalent of about 300 to 600 full length novels. Access to any particular program or set of data on a disk pack is perhaps 1/10th of a second or less.

There are many other secondary storage devices. The idea is to store large amounts of information cheaply and reliably. Gradual progress is occurring on a year by year basis, with an occasional new idea every few years. Of particular interest to users of inexpensive computer systems is progress in developing floppy disks. A floppy disk is a thin flexible plastic disk coated with iron oxide material. An eight-inch diameter disk, costing about $5, can store a half million bytes. A floppy disk drive, which reads and write 8-inch floppy disks, costs about a thousand dollars.

A minifloppy disk is now quite common. It is 5¼ inches in diameter and costs under $4 per disk. A minifloppy disk drive costs about $500, and one disk stores about 100 thousand to 200 thousand bytes.

Floppy disks are cheap and relatively reliable. They are much more reliable than cassette tapes, and much more convenient to use. However, they are less reliable and have less storage capacity than the larger hard disks discussed earlier.

Another secondary storage device that will eventually be common in microcomputers is called bubble memory. Bubble memory is manufactured using many of the same ideas used to produce the large scale integrated circuitry for a CPU or primary memory. However, it is permanent memory, like a magnetic tape or disk. Of course it can be erased, so that it can be used over and over again. Bubble memory is highly reliable, since it has no moving parts. Right now it is more expensive than floppy disk secondary storage, but rapid decreases in cost are occurring. It seems likely that many microcomputers of the future will contain both bubble memory and floppy disk memory for secondary storage.

Applications

Today's children will be adults in a world in which entire libraries are stored in computer secondary storage, and in which computerized information storage and retrieval is commonplace. The activities listed below are designed to help students understand the large numbers and large quantities of data that computerized information storage and retrieval involve.

1. How much is a million?
   a. Examine various dictionaries. How many words are defined in each dictionary?
   b. Examine a very thick (adult) novel. Does it contain a million characters?
   c. Examine the telephone book for your city and/or for a very large city. Does it contain a million names and phone numbers?
   d. How many books are in your school library? Is this a million pages?
   e. Select a table or desk in your classroom. Suppose it were evenly covered with pennies, stacked high enough to hold a million pennies. How high will the stack be?
   f. How many seconds are there in a year? Is it more than a million?
   g. How many people live in your state? Is it more than a million?

h. Is the earth more than a million meters in circumference?

i. How long would it take you to write down all of the numbers 1, 2, 3, and so on, counting up to a million? Make an estimate based upon starting at 100,001 and seeing how far you get in a minute. Suppose that a line printer is printing 20 numbers per line and 1,000 lines per minute. How long will it take this computer driven printer to print out the numbers from 1 to 1,000,000?

Exercises

1. A 2400 foot reel of tape is recorded at a density of 800 bytes per inch. How many characters can it store? Solve the same problem for recording densities of 1600 bytes per inch and 6250 bytes per inch.

2. A very high speed tape drive has a speed of 125 inches per second. How long will it take this tape drive to read a 2400 foot reel of tape?

3. An 8-inch floppy disk costs $5 and stores a half million bytes. A large disk pack costs $300 and stores 300 million bytes. What is the cost per byte of storage for each of these disks?

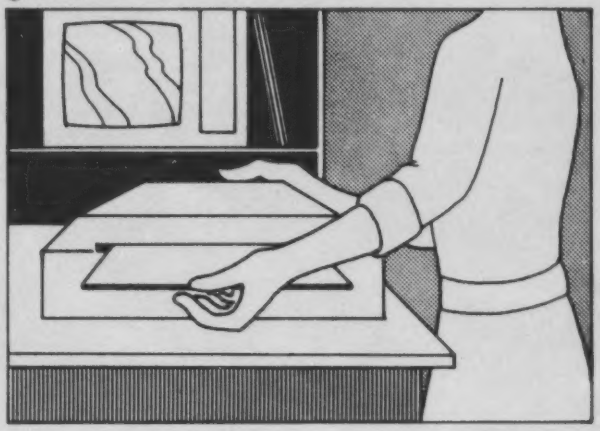
4. A medium-sized school district has 10,000 students. For each student, the district maintains an extensive set of data such as student and parents' names and addresses, academic record and attendance record. Discuss the feasibility of using a microcomputer with an 8-inch floppy disk drive to store and process this data.

5. A certain small college's library contains about a hundred thousand books. The college president suggests storing the contents of all of these books on large disk packs, providing access via computer. Discuss the feasibility of this project. Take into consideration that a disk drive for a large disk pack costs $15,000 or more.

6. A 2400 foot reel of tape, recorded at a density of 6250 bytes per inch, is being printed on a line printer that prints 132 character lines at the rate of 1000 lines per minute. How long will it take to print the entire tape?

Videodisk

The video disk is primarily a product of the television industry. The goal was to develop a method better than videotape for the storage and retrieval of television programs.



The results have been spectacular! One type of videodisk system uses a laser light beam to read a rapidly spinning plastic disk. One side of a disk contains 54,000 individual pictures (frames), the equivalent of a half hour of television. These disks can be produced cheaply and rapidly, using a stamping process similar to that used on phonograph records.

This video disk player can be hooked to a microcomputer. Under computer control any single frame can be accessed and displayed. Forward and backward motion, fast and slow motion—all are possible. The videodisk system is equipped with two sound tracks, which can be useful in bilingual education or in reaching different types of audiences.

The computer-controlled videodisk system is currently (1980-81) still under development. A number of educational research projects are developing demonstration instructional materials and exploring the effects of using these materials in education. Materials are being developed for special education, general education, military education and for industrial and business education. Over the next five to ten years, the videodisk will come into millions of homes. Computerized videodisk systems will begin to make a significant contribution to education.

CHAPTER THREE

AUTOMATIC FLASHCARDS

Instructional use of computers can be divided into three major parts:

1. Teaching about computers.
2. Impact of computers on the curriculum.
3. Teaching using computers.

The goal in the first case is increased knowledge about the general capabilities, limitations and applications of computers. The emphasis is upon learning to use computers as an aid to problem solving, and developing a useful level of knowledge about computer science. At the college level, such instruction is often provided in a Computer and Information Science Department or a Computer Science Department. Some high schools are beginning to think about having a Computer Science Department. At the elementary and middle school levels, such instruction can conveniently be integrated into various other departments such as language arts, social studies, math, science or the library/media center. The first two chapters of this book are examples of teaching about computers.

Computers have the potential to have a major impact upon curricular content. Chapters Five and Six discuss and illustrate some possible changes in the current curriculum that may occur as computers become more readily available.

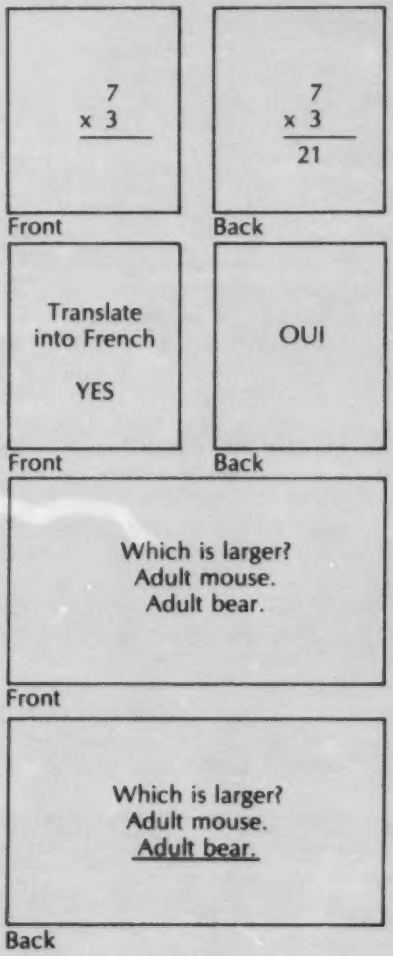
This chapter is the first of two discussing teaching using computers. Computers are a new educational medium, with a tremendous potential to impact the instructional process. Their use as an instructional aid is now well established and is growing rapidly.

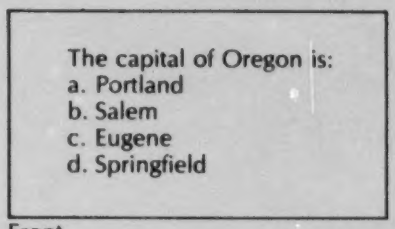
Teaching using computers is often called computer assisted learning (CAL), or sometimes referred to as computer assisted instruction (CAI). The goal in CAL is increased knowledge of some subject matter area (which might happen to be computer science, but could just as well be language arts or music). CAL is often divided into a number of categories such as drill and practice, games and simulations, problem solving or dialogue. The most commonly used mode of CAL is drill and practice. Here the computer is used as an automated flashcard machine. This chapter is devoted to various aspects of computerized drill and practice and serves as a general introduction to many of the key ideas of CAL.

Elementary school students are expected to memorize many facts, such as the one-digit addition facts, and to spell correctly a large number of different words. They are expected to develop skill in the use of these facts to accomplish more complicated tasks. One-digit addition facts are used in summing a column of numbers, while spelling is used in writing.

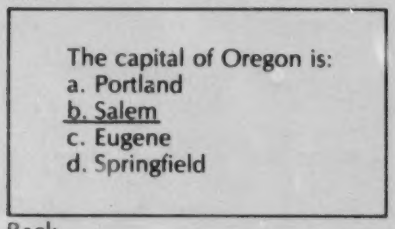
Learning theory and practical teaching experience tell us quite a bit about rote memory and about developing skill at more complicated tasks. While students differ considerably in their learning and retention rates, almost all students can memorize a great deal of material and can develop considerable skill in applying this knowledge. An important aspect of this type of learning is drill and practice, with appropriate feedback.

One way to provide such drill and practice is via flashcards. Young children may need to have the flashcards presented by an adult or another student, while older students can handle both the presentation and the feedback (looking at the answer) for themselves. A wide variety of materials can be presented via flashcards.





Front



Back

Teachers are continually looking for ways to make drill and practice more interesting and fun. One can purchase books of pictures, cartoons or jokes interspersed with rote memory material to be drilled upon. Thousands of games and other learning aids have been developed. Almost every elementary and middle school teacher has developed and/or used such materials, and most teachers eagerly embrace clever new ideas.

Rote memory is an important aspect of learning, but it is by no means the whole answer. For example, consider the mathematics education of very young students. They need to work with manipulatives, to have lots of hands-on experiences with things they can touch, move around, count, rearrange, and so on. This concrete phase of learning is essential to their ultimate development of more abstract skills.

Equally important, as students begin to deal with more complex problems, is strategy. How does one attack a problem? A student may memorize that five plus five is ten. Can the student use this, along with simple counting, to mentally figure out that five plus six is eleven? With a little help a child can learn a strategy. "I know that five plus five is ten. Six is one more than five. So five plus six must be one more than five plus five. One more than ten is eleven. The answer must be eleven." Strategies are applicable to a wide range of problems and are an essential part of a quality education. Rote memory is not a good substitute for a combination of understanding and being able to "figure it out."

The use of a computer for drill and practice is by far the most common instructional use of computers in elementary and middle school education. Very extensive research over the past 20 years has shown that computerized drill and practice is an educationally effective aid to learning. By and large, however, it has not proven to be sufficiently more effective than other aids to instruction to justify the high cost.

Indeed, high cost has been a major barrier to CAL at all educational levels. The most extensive uses of CAL have been in medical education, certain aspects of military education, special education, and other places where conventional education is quite expensive. But the cost of computer facilities has declined drastically over the years, and continues to do so in terms of real (constant value) dollars. Now, in many educational settings, a computer is both a cost effective and an educationally effective aid. The computer is not a replacement for the other aids that have served education so well, but is certainly a useful supplement.

Exercises

1. Give at least five examples of methods and/or aids useful in drilling students on each of the following topics.

   a. Arithmetic facts
   b. Spelling
   c. Geography facts
   d. History facts

2. What number facts (for example, the one-digit times table) are students expected to have memorized by the end of the sixth grade? How many facts is this? How many words do you think the average sixth grader can spell correctly? Give evidence to support your answers.

3. Monopoly is a long-lived and widely played game. The players traverse a board, buying and selling property while constructing houses and hotels. Money continually changes hands.

   a. What types of things might students learn by playing Monopoly or a similar game?
   b. Discuss the merits and feasibility of allowing students to play Monopoly or a similar game in school during regular in-class hours.

   A question such as this is quite relevant to computers, since there are many interesting, exciting and educationally relevant games available for use on computers. Many of these games are designed to drill students on facts relevant to their overall education. We will discuss the idea of computerized games in the next chapter.

4. Does being able to correctly spell a large number of words make a person a good writer? Does being fast and accurate on memorized number facts make a person good at solving math problems? Discuss the idea of memorized "facts" versus learning to think and to create using the facts. Give your opinion as to the percentage of school time that *is* spent and that *should be* spent in each of these areas.

A Simple Model

The essence of drill and practice is presented in the flowchart of Figure 1. A flowchart is a collection of

boxes connected by arrows, designed to specify a set of directions. Flowcharts are so commonly used by computer scientists that they have agreed to assign meaning to different shapes of boxes. Thus, the START and STOP boxes are eliptical shapes, while a diamond is used for decision making. For simplicity, we use a rectangular box for all other purposes, although computer scientists use additional shapes.

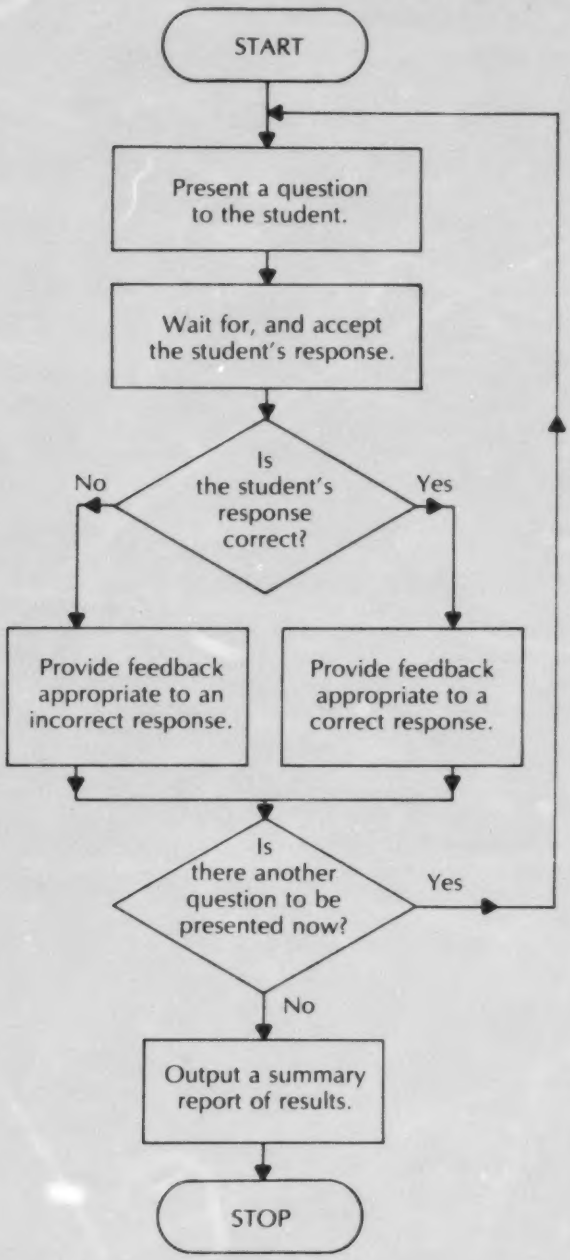

START
Present a question to the student.
Wait for, and accept the student's response.
Is the student's response correct?
No
Yes
Provide feedback appropriate to an incorrect response.
Provide feedback appropriate to a correct response.
Is there another question to be presented now?
Yes
No
Output a summary report of results.
STOP

Figure 1

This flowchart does not provide adequate directions to a computer programmer who wants to write a drill and practice program. It does not give the source or nature of the questions, the type of feedback to provide in the case of right or wrong answers or the number of questions to be asked. These are issues that should be decided by an educator, not just a computer programmer. (Of course the programmer may be an educator.) The next few sections will provide more detailed information on options available to the program designer.

Good educational software is often developed by a team. The team may include a teacher or subject matter expert, a learning theorist, a graphic artist and a programmer. Sometimes a single person will fill two or more of these roles. You should be aware that writing good quality computerized educational material is a major task. It is more difficult than writing good quality textbook materials; thus, it is easy to understand why there is a scarcity of good CAL materials, and why such materials tend to be expensive.

Where Do the Questions Come From?

In computerized drill and practice, the computer must have access to a large supply of questions or must have some method for generating questions. Both approaches are commonly and successfully used.

The idea of a data bank of questions, residing in secondary storage, is simple. Instead of constructing flashcards out of paper or cardboard, one inputs the questions and answers to computer memory. For example, suppose you want students to memorize the capital of each state. The data bank might contain a list of the states and their capitals.

The creation of such a data bank is straightforward. A computer program is written to accept as input questions and their answers. A teacher or other educator inputs each question and answer, and the program stores these on a magnetic disk or other secondary storage medium. Later, when a student wants to drill on these questions, a computer program selects questions from the data bank and presents them. The computer compares the student's answer with the answer that has been stored in the data bank and provides appropriate feedback.

An important alternative to data bank-based drill and practice is the use of computer-generated questions. The simplest examples of generative drill and practice come from arithmetic, and make use of randomly generated numbers. It is relatively simple to develop a computer program that generates random numbers in a specified range. Many games and educational devices make use of random numbers generated by throwing dice, spinning a spinner or drawing numbers out of a hat. The computer generation of random numbers is a little more sophisticated, but produces equivalent results.

Suppose one wants to drill students on one-digit addition facts. A computer can easily generate such ques-

tions, along with correct answers. The flowchart given in Figure 2 illustrates the essential ideas. We have used algebraic notation: A for the first number, B for the second number, S for their sum. But this is merely for our communication convenience. If, at a particular time, the computer generates values A = 7 and B = 5, the student would be asked to find the sum of 7 and 5.

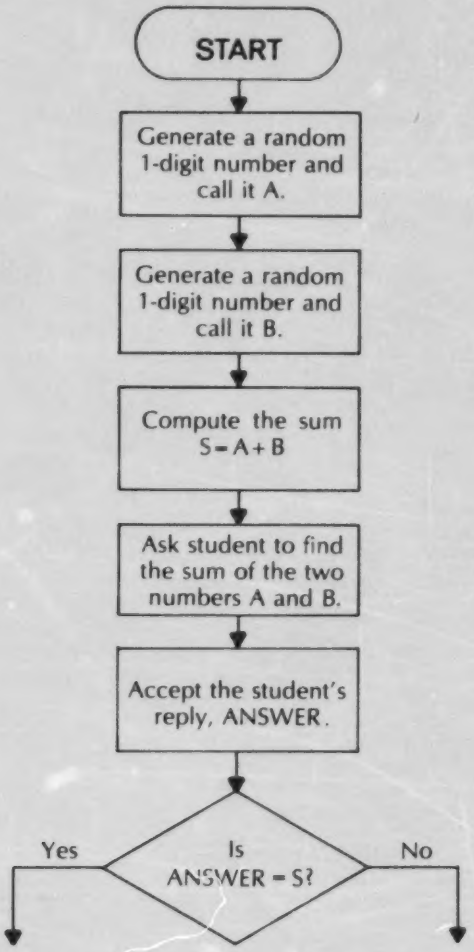
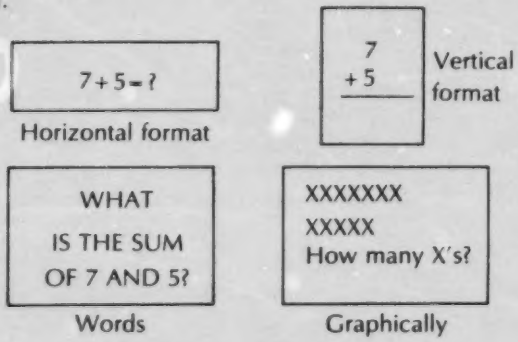

START
Generate a random 1-digit number and call it A.
Generate a random 1-digit number and call it B.
Compute the sum S = A + B
Ask student to find the sum of the two numbers A and B.
Accept the student's reply, ANSWER.
Is ANSWER = S?
Yes
No

Figure 2. Generation of 1-digit addition fact questions.

There are many ways to actually present such a question.



It is straightforward to use any one or any combination of these presentation formats. The computer can be programmed to randomly select the format to be used, or to cycle through a collection of formats.

Data bank-based drill and practice can require a substantial amount of secondary storage and considerable effort on the part of educators creating the data bank. If the data bank is small, the students will soon notice the exact same questions occurring over and over again. Generative drill and practice, on the other hand, requires a more sophisticated computer program, but generally uses less secondary storage. It takes careful thinking and planning on the part of the educators and programmers creating the program. Thus, the creation of a generative drill and practice program tends to be more intellectually stimulating and requires less routine work than the creation of a large data bank of questions.

Applications

The idea of random numbers or random events is important, not only with computers but to all of science. Activities given below are designed to help increase student insight into random numbers, and are computer readiness activities. They can be used in a math or science unit. Organizing and recording results are an important part of these learning activities, and are an important part of math and science. Results may be represented in various forms such as a table, graph, chart or histogram. Computer output in all of these forms is also possible.

1. Have each student roll a die a number of times and record the outcomes. Does one number come up much more often than others? Combine the results for the whole class on the chalkboard. Does one number come up much more often than others? Repeat the entire exercise and answer the same questions.

   If the dice are all "perfect," then one can expect that no single outcome will dominate when the dice are tossed many times. But in small experiments and/or with non-perfect dice, one may well find a single outcome occurring a disproportionate amount of the time.

2. What is the most likely sum produced by throwing two dice? The possible outcomes from one throw are the sums 2, 3, 4, . . . , 12, and these are not equally likely. First have each student guess the most frequently occurring sum. Then have each student throw a pair of dice a number of times, and record the outcomes. Finally, combine the students' outcomes on the chalkboard and have a class discussion on the results. The same activity can be used with the product of the numbers appearing when a pair of dice are thrown. Or, it can be done using three or more dice. Notice that this activity gives practice in working with simple arithmetic facts as well as

contributing to the other goals listed at the beginning of this Applications section. Too often math drill and practice is dull and devoid of application.

3. Give each student five coins or five similar two-sided objects. Decide which side is "head" and which side is "tail." The students are to toss their coins a number of times, recording the number of heads and number of tails resulting each time. For example:

Toss Number	Heads	Tails
1	4	1
2	3	2
3	4	1
4	0	4

What was the most frequent outcome? What was the least frequent? Use the chalkboard to combine the students' results, and discuss. Repeat the entire experiment and see if the results are approximately the same.

This same experiment can be run with different numbers of coins. Students will learn that certain outcomes, such as all heads or all tails, are infrequent. They will learn that the outcomes with nearly equal numbers of heads and tails will be most frequent.

4. Any of the above activities can be repeated for other devices with random outcomes. Thus, one can use spinners, numbers in a hat or four-sided dice to generate random outcomes with certain characteristics. Computer programs can be written to simulate these experiments. If you have access to a computer and this software, you may want to run such computer simulations and compare the outcome with the student-generated results. This use of computer simulation is one way to introduce the idea of computer simulation to a class. The computer carries out very rapidly and accurately certain aspects of an activity such as tossing a die. The computer simulation does the tossing and accumulates the outcomes. But the computer simulation does not include the fun of children interacting or the laughter as dice fall on the floor. A simulation focuses only on certain specified aspects of the situation being simulated.

Exercises

1. Make a list of things that students are expected to learn via rote memory in elementary and middle school. Be as succinct as possible, but try to include as many major categories as you can. (Refer back to the first set of exercises for this chapter.) Which categories seem to readily lend themselves to computer-generated (that is, generative) drill and practice questions? For each category you name, describe a method whereby a computer program could generate appropriate questions.

2. A certain computer program draws a map of the United States on its graphics terminal, labeling each state with a different number. The student selects which of four options s/he wants to try:

   Option 1: The computer displays the name of a state; the student types in the state's number.

   Option 2: The computer displays a state's number; the student types in the state's name.

   Option 3: The computer displays a state's name and number; the student types in the capital.

   Option 4: The computer displays a state's capital; the student types in the state's name.

   In all cases, the program provides immediate feedback of

   YES, THAT IS CORRECT

   for a correct answer, or

   NO, THAT IS INCORRECT.
   THE CORRECT ANSWER IS .....

   and supplies the correct answer when the student errs.

   a. In what ways is this like a data bank-based drill and practice program and in what ways is it like a generative drill and practice program?

   b. What aspects of this program seem to be especially useful or good, and what aspects seem educationally poor or unsound?

   c. Give an example of some other body of knowledge that might be taught in elementary or middle school by a program with somewhat similar computer capabilities and program characteristics. Justify your answer.

3. An audio output device can be used by a computer to present spelling words. The student keys in the correct spelling. Texas Instruments Company sells a machine for about $55-60, called the Speak and Spell, which accomplishes a similar task.

   a. Discuss the merits of this type of drill and practice as an aid to improved spelling.

   b. Suggest at least two different ways this computer program could be improved (expanded) to be an even more useful language arts learning aid.

4. Examine the flowchart of Figure 3. It presents an abbreviated outline of a generative drill and practice program. If a student gives a wrong answer, the same question is presented again. Discuss the educational merits of this drill and practice routine. Include in your discussion both good and bad features as well as your feelings on how well this drill would work with students.

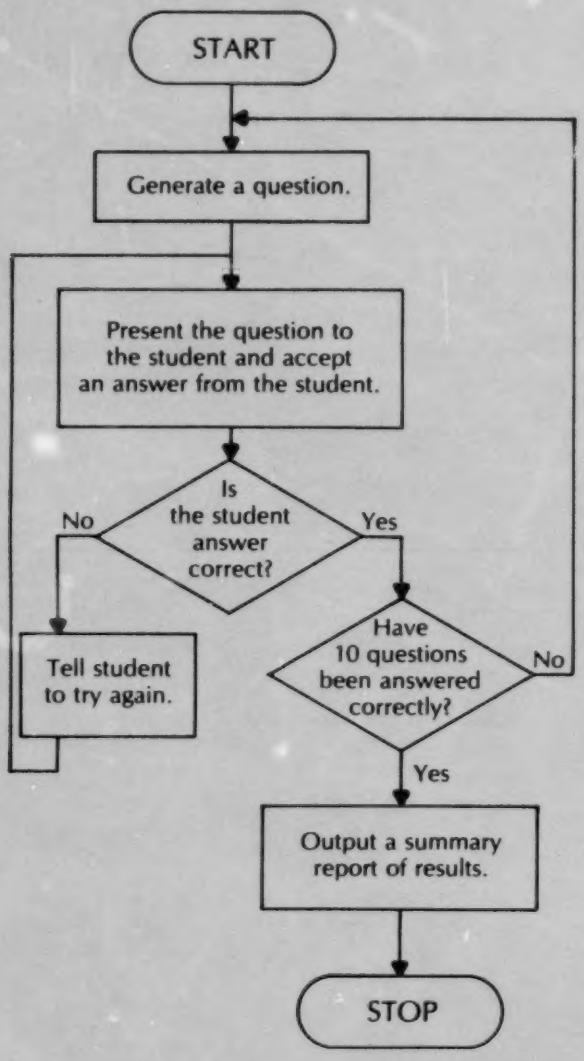
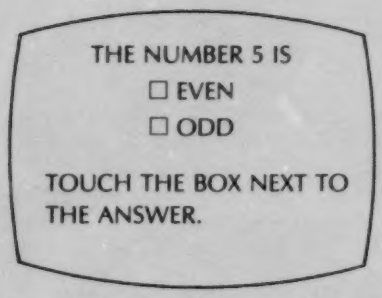
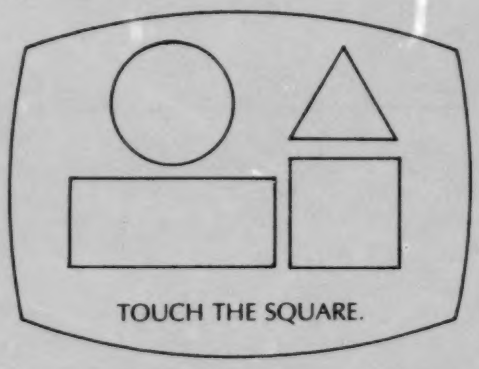


Figure 3.

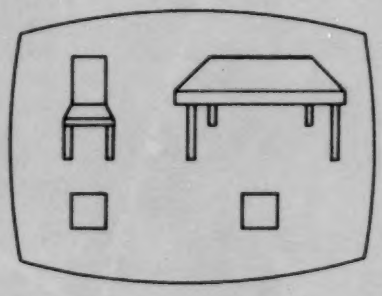


Here the student touches the box to the left of the correct answer. (Notice that this ODD-EVEN question is easily programmed in a generative CAL mode, using a random number generator.)

Touch input is quite versatile.



Such computerized drill can be used with non-readers. The directions are "spoken" by the computer via computer audio output device. The spoken directions might be TOUCH THE BOX BELOW THE CHAIR.



The Student's Response

With ordinary flashcards, a student's response is usually given verbally to a person who interprets it and then provides appropriate feedback. That is, a human being receives and processes the answer.

With computerized drill and practice, there are many possible methods to input responses. The most common is via keyboard terminal, with a student typing an answer. Later in this section we will discuss this in more detail. A useful alternative is via touch panel, or touch sensitive screen. The student is asked to touch the correct response.

Programs such as this can be used with pre-school children and with learning-disabled children to build language arts skills.

Verbal input to computers is now fairly well established, but is still in its infancy. It is likely that you have seen television programs or newscasts showing a voice controlled wheel chair. A quadriplegic can direct the wheel chair by voice commands such as LEFT, RIGHT, STOP. Similar computer equipment can be used to allow students to respond to a computerized drill and practice program.

A voice recognition system works with digitized (numerical) representations of the words to be recognized. The system is "trained" to recognize a certain set of words spoken by a certain person. In a typical system, the user speaks each word several times. The computer analyzes the digitized input and stores away a mathematical pattern for each word. Later, when the system is asked to recognize a spoken word, it does so by comparing a digitized representation of the word against each of its stored patterns.

An inexpensive word recognition system, costing under $500, may be able to store only a few dozen words. Moreover, its recognition error rate may be quite high (several percent). Thus, such a system is most likely to be used as an interesting experimental tool or as a toy. More expensive systems, costing perhaps $5,000 to $20,000 or more, allow for larger vocabularies and have much smaller recognition error rates. These are reliable enough to be used in special education, as input devices for cash registers in a super market or to control wheelchairs and hospital beds.

The problem of handling a large vocabulary in the form of words strung together in phrases or sentences is still a research topic. This is called connected speech; it is the type of speech one uses in ordinary conversation. The words in connected speech are run together, rather than being distinct and carefully pronounced. It will be many years before a connected speech recognition system is commercially available at a reasonable price. The voicewriter (voice input typewriter) is not likely to be a standard commercially available item during the next ten years.

As mentioned at the beginning of this section, the cheapest and most widely used student response mode for computerized drill and practice is via keyboard terminal. This poses a specific problem—most young students have poor typing skills. "Hunt and peck" is mainly hunt; students' responses are quite slow, so that the computer terminal is tied up for a long period of time. Very young students can be taught touch typing and there are some people who advocate doing so. Indeed, there has been considerable research on the use of a computer to teach touch typing. Such research and/or the use of a computer to teach touch typing to elementary and middle school students has not been widely accepted or implemented, even though such students are quite capable of developing good typing skills.

Exercises

1. Many young children learn to play a piano and/or other musical instruments which involve considerable manual dexterity, finger strength and hand/eye/ear coordination. It is evident that these children could learn to type quite well. Currently, however, typing is mainly a junior high school or senior high school course. Why do you think this is the case? Discuss whether the introduction of computers into elementary and middle schools could/should/will change this.

2. Not only can young students learn to type, they can learn to use a word processing system. A word processing system is a computerized typewriter. Preliminary indications show that students enjoy using a word processing system and that it contributes to their creative writing skills.

   However, when students are using a typewriter or word processor they are not practicing their handwriting skills. Present and discuss your opinions on introducing typewriters and word processing into the elementary school. Do you feel qualified to decide whether this is appropriate, and do you feel qualified to teach the use of these tools?

Response Processing

Another problem in computerized drill and practice is response processing—receiving student answers and figuring out if they are correct. Suppose that a computer presents the question:

WHO WAS THE FIRST PRESIDENT
OF THE UNITED STATES?

What should the computer do if a student responds WASHINGTON? Should the program expect that GEORGE WASHINGTON (first and last name, both correctly spelled) is the only acceptable answer? Suppose the student types WASHINGTON, GEORGE. Is this acceptable? How about HE WAS THE FATHER OF OUR COUNTRY?

You can see that this is an immensely complex and difficult problem. A human can receive an answer, interpret it, and try to figure out what was intended. A human can explain what s/he understands as the proposed answer and request clarification. To write a computer program with similar processing capabilities is very difficult. Many aspects of it are still unsolved research problems.

There are two commonly used simple solutions to the response processing problem. The first is to provide students with a list of alternative answers and ask them to select one.

WHO WAS THE FIRST PRESIDENT OF THE UNITED STATES?

1. THOMAS JEFFERSON
2. BOOKER T. WASHINGTON
3. JOHN ADAMS
4. GEORGE WASHINGTON
5. MARTIN LUTHER KING

Now the student needs only to key in the number of the correct answer. But this restricted response list type of question does not teach the same things as the open response format. The "real world" does not consist of multiple choice questions, with a single correct answer always being among the available choices!

A second alternative is for the educator creating the computer drill questions to provide an exhaustive list of answers that are to be recognized by the program, along with the action the computer is to take in each case. Generally this means the program is quite rigid. The first president question might have as acceptable answers:

GEORGE WASHINGTON
WASHINGTON
GEORGE

The computer's response to the first answer might be GOOD, THAT IS CORRECT. The computer's response to the second answer might be WHAT WAS HIS FIRST NAME? The computer's response to the third answer might be WHAT WAS HIS LAST NAME? For any other response, the program might respond THAT IS NOT CORRECT. The smallest typing or spelling error would produce the latter response. HE WAS THE GENERAL WHO LED OUR ARMY IN THE REVOLUTIONARY WAR would elicit the same computer response as MICKEY MOUSE.

Most computerized drill and practice programs currently available use one of the two alternative methods of response processing listed above. But progress is occurring in developing high quality response processors. For example, programs have been written that are able to detect when two words are nearly the same, so that one may be a misspelling of the other. Programs have been written to take a student's response and analyze it and to look for parts that may be related to a correct answer. Such a program might accept an answer such as GEORGEEWASHINGTON and respond in a suitable fashion.

Applications

1. Show your class examples of flashcards. Discuss with them the idea of different presentation formats and different formats for answers. Then have each student make a set of flashcards for something that s/he wants to learn. Let the students practice using these flashcards on each other.

   This is an important "learning to learn" activity. Through activities such as this, students can learn how to help themselves learn, and can begin to take more responsibility for their own learning.

Exercises

1. Select a small body of knowledge that is suited to computerized drill and practice. The exercises a. and b. given below can be done with pencil and paper. Or, if you are a reasonably good programmer, use a computer.

   a. Create material suitable for use on a computer that uses the multiple choice format to insure that the student's response can be understood by the computer.

   b. For the same body of knowledge, create material that uses an "exhaustive list of answers," and what action you want the computer to take for each of the possible answers.

   c. Finally, develop a small amount of material that requests an open ended response. Discuss the variety of types of responses that one might expect, and the difficulties in developing a computer program to handle these responses.

   Discuss which of the answer formats is best suited to young students learning your material.

2. Children enjoy playing games in which the objective is to guess an object in the room, or perhaps an object or event being thought of. The person running the game provides clues such as "hotter" or "colder." These mean the guesser is getting closer to the answer, or further away. Make up an example of an educational guessing game in which a computer could process the player's responses to provide "hotter" and "colder" feedback.

Feedback

At first glance, the question of feedback seems simple. For each question, the computer accepts a student's response and reports to the student whether it is correct. After a drill session is completed, the computer outputs a final report summarizing the session.

Feedback for a correct student response might consist of the printed message THAT IS CORRECT. But this message, received over and over again, is rather boring. How about a smiley face, a merry tune, a ringing bell, a variety of different written messages (GREAT!; RIGHT; YOU ARE DOING WELL; KEEP UP THE GOOD WORK; CORRECT; GOOD; and so on)? If desired, one can program a computer to print out a coupon that can be exchanged for a stick of sugarless gum or other type of encouragement for correct answers.

What about an incorrect response? There are many ways to tell a student s/he has made a mistake. Should the student be given a second chance, or a third chance? Should the student be given a helpful hint? If so, then the response processor part of the program may need to be quite sophisticated in order to determine the student's source of difficulty.

The feedback used in some of the software currently available commercially is quite poor. One can find examples in which the computer response to an incorrect answer is much more entertaining than the response to a correct answer. Students deliberately make errors to receive the more entertaining response. In other cases, the response to errors is demeaning, overly critical and harsh, all of which can be damaging to the learning process.

Part of the overall feedback mechanism is the reinforcement schedule. When a student gives a correct re-

sponse, how soon should the same question be asked again? When a student gets a question wrong, how soon should the question be asked again?

Research into learning theory is providing some help on feedback mechanisms and reinforcement schedules. But it is clear that there is considerable room for progress, both in the theory and in its computerized implementation. For example, a standard assertion is that computerized drill and practice is good because the computer can provide immediate feedback. Recent learning theory research suggests that people have both a short term memory and a long term memory. Immediate feedback may have less impact upon long term memory than delayed feedback. That is, a student may actually learn more if feedback is delayed for a period of time, such as 15 seconds! It is evident that more research is needed. It may well be that differences between people are so great that feedback adjusted to fit the needs of one student would be totally inappropriate for another student in the same class.

It is easy to program a computer to keep track of the number of questions attempted, the number of correct responses, the number of incorrect responses and the elapsed time. This information may be used during the drill to adjust the level of difficulty of the questions being presented. It can be used at the end of the drill to provide information to the student or the teacher, and perhaps to add information to the student's file, which may be computerized. In the latter case, during a drill session the computer can use information accumulated from previous sessions with the student.

Applications

1. Talk to your students and find out what kind of feedback and what type of reward system they like. For example, if a student spells every word on a list correctly in three minutes or less, perhaps the student should be allowed to play with a calculator or computer for five minutes. Have each student express how s/he would like to be corrected in the event of an error, and what sort of reward would be preferred for good performance.

2. If you have access to a computer in an elementary or middle school classroom, have students use various drill and practice programs. Each student is to write a brief report on which program s/he liked best, and why. Suggest that they pay particular attention to the feedback provided by the programs, and that they comment on these features.

Exercises

1. What type of feedback do students get in an ordinary elementary school classroom? How much is provided by (a) teacher or aide, (b) self (perhaps via answer book), (c) other students? How much of these various types of feedback could be provided by a computer?

2. How much flashcard-type drill per student per day seems to you to be appropriate for elementary and middle school students? Does this vary with the grade level or time of year? Suppose that an elementary school has two classes of 25 students in each of the grade levels 1-6. That is, the school has 12 classes. Suppose further that classes are in session six hours per day. How many microcomputers or computer terminals would such a school need to provide the amount of flashcard-type drill you think is appropriate?

3. A computer program designed to teach the one-digit addition facts to first and second graders ends with the following report to a student.

   PROBLEMS ATTEMPTED 20
   CORRECT RESPONSES 16 = 80%
   INCORRECT RESPONSES 4 = 20%
   YOUR GRADE, TERRY, IS B.

   Discuss the suitability of this summary, and ways to improve it.

4. Examine the current literature on feedback as it relates to student learning, or review your knowledge gained in a learning theory course taken as part of your education program. Then, run several computerized drill and practice programs, paying particular attention to the feedback provided to the user. Write a report comparing learning "theory" with the actual practices demonstrated in the programs.

5. Many computerized drill and practice programs make use of pictures or cartoon figures, which may be in color and be quite impressive, for feedback. Such programs may also use sound, such as assorted beeps, or even music. Are you aware of published research results that such computerized feedback increases student learning? Discuss this situation.

Final Remarks

Many parents and teachers yearn for the good old days, when life and education were simpler. There was no television, and the "work hard" ethic was stronger. Students were expected to learn the basics of reading, writing and arithmetic, and the educational system was considered a success if this occurred.

Times, however, have changed, and education cannot be separated from the environment in which students live and grow up. It is a fact of life that the average 18-year-old has spent more hours watching television than attending school. The computerized home entertainment center is beginning to add to this problem. Students in the future will grow up having access to computerized games that are even more fun

than television. They will have had hundreds of hours of such computer experience before they get to the first grade.

This problem is further compounded by the fact that computer assisted learning materials will be available for use in the home. Students will enter the first grade with a still wider range of backgrounds and skills. As computers come into the elementary and middle school, and teachers struggle to cope with this new learning/teaching medium, they will also have to cope with students for whom the computer is old hat.

There is no simple answer to these problems, but teacher knowledge is certainly part of the key. If you are familiar with computers as an educational medium and with the computerized home entertainment center, then you will have a better chance to cope with children growing up in our computerized society. You will be able to build upon this new technology as it becomes available, first in students' homes and later in your classroom. Computers do contribute to the complexity of life in our society and in education; however, they can also help students in learning to cope with this complexity.

CHAPTER FOUR

EDUCATIONAL GAMES COMPUTERS PLAY

You are aware that there are many computerized games. The old-fashioned pinball machine has been computerized and supplemented by electronic space war, electronic car or horse races, electronic football, and so on. The home entertainment center video games could initially play "Pong" and other simple paddle games. Now a wide variety of sophisticated interactive games are available. The leading edge of the interactive computerized game market is the arcade-type games available at commercial entertainment centers. But similar games, in much greater variety, are available for home computers. Games that take a long time to play such as "Adventure" games, the "Dungeons and Dragons" games, chess and checkers are specifically designed for the home computer market. It may take hours to play one of these games.

Most games computers play are designed for entertainment purposes; however, some of these programs also have redeeming educational value. An examination of ads for microcomputer software suggests there is far more entertainment software than instructional software available at the current time. Thus, an elementary or middle school wishing to make instructional use of microcomputers is apt to make use of some of the entertainment software. There is a rapidly growing collection of software, however, that is both entertaining and of educational merit. We will examine several games that have significant educational value.

GUESS

Computer games usually have names. We have named the one to be discussed in this section GUESS. GUESS is a simple number guessing game. Suppose Terry has just called this program out of a library and provided the computer with his/her name. The computer responds:

```
HI TERRY. I AM HAPPY THAT YOU
WANT TO PLAY A GAME OF GUESS
WITH ME. I AM THINKING OF A
NUMBER BETWEEN 1 AND 25. SEE IF
YOU CAN GUESS IT. I WILL GIVE YOU A
HINT IF YOU MISS. WHEN YOU ARE
READY TO BEGIN PUSH THE ENTER KEY.
?
```

Often a game program is written by a programmer for his/her own entertainment, or possibly for use by some larger audience. We see an immediate problem. Suppose you want to use GUESS with second grade students. Can they read and understand the directions?

Typically this will be a serious problem. The game program author may not be an educator, and may not write clear, concise directions understandable by students at a particular grade level. For example, can the "answer" in this game be 1, or 25, or 16.8? Most likely the game author intends a whole number in the range of 1 to 25 inclusive, but the directions do not make this clear. A teacher using the program will need to experiment with it and may need to explain it to his/her students.

Terry reads the directions and pushes the ENTER key. The screen is cleared and then:

```
OK TERRY. WHAT IS YOUR FIRST GUESS?
```

Terry types 16 and nothing happens. After a while Terry types the ENTER key. (The directions are poor; the ENTER key must be typed after each guess.)

```
OK TERRY. WHAT IS YOUR FIRST GUESS?
16
THAT IS TOO HIGH. GUESS #2?
```

The guess of 16 was TOO HIGH. Terry may interpret the message "GUESS #2" to mean that s/he should type in a 2. Again we see that the machine-user interaction is poor. Let us suppose that Terry continues with a lower guess, and eventually guesses the computer's number. The final screen display is as follows:

```
OK TERRY, WHAT IS YOUR FIRST GUESS?
16
THAT IS TOO HIGH. GUESS #2? 12
THAT IS TOO HIGH. GUESS #3? 5
THAT IS TOO LOW. GUESS #4? 9
THAT IS TOO LOW. GUESS #5? 10
RIGHT!! YOU GOT IT IN 5 GUESSES.
DO YOU WANT TO PLAY AGAIN (YES,
NO)?
```

Typically a game continues until the student guesses the correct answer or until the number of guesses exceeds some set limit. The student can play the game over and over again, with the computer generating a random integer in the range of 1-25 inclusive each time.

What is the educational value of this game? Some possible answers include:

1. Reading skills. The student must read and follow directions.
2. Number skills. The student must understand the meaning of TOO HIGH or TOO LOW. The student must be able to figure out guesses in the appropriate range.
3. Problem-solving strategies. With a correct strategy a student can always guess the computer's number in five tries or less. A student may discover this strategy by trial and error or by analysis of the problem. Alternatively, the teacher can teach the strategy.
4. Computer awareness/familiarity. The student uses a computer and develops confidence in his/her ability to cope with this machine.

Whether these educational values are realized by a student or a class depends mainly upon the teacher. If the students are turned loose on a computer with little supervision or instruction they will have fun. With games like GUESS, some incidental learning will occur, but it will probably be quite limited. For example, a student may learn the rules by being "told" by another student, rather than by reading. Or, consider the following situation.

```
OK TERRY. WHAT IS YOUR FIRST GUESS?
16
THAT IS TOO HIGH. GUESS #2?  23
THAT IS TOO HIGH. GUESS #3?  28
```

A teacher might analyze these responses and see that the student does not understand the meaning of TOO HIGH, and also does not understand that the answer must lie between 1 and 25. Some remedial instruction is needed. This remedial instruction could be built into the program, and it will be if the game has been designed to be a good teaching aid.

```
OK TERRY. WHAT IS YOUR FIRST GUESS?
16
THAT IS TOO HIGH. GUESS #2?  23
THAT IS A POOR GUESS. YOU HAVE
BEEN TOLD THE ANSWER IS LESS THAN
16. TRY AGAIN. GUESS #3?
```

If a student plays a sequence of games and continually takes more than five guesses, the teacher or the program might suggest that five or less guesses will always suffice. Some students will discover this by themselves. A well-written program can teach it.

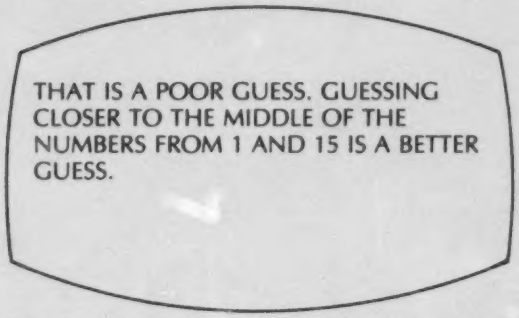
```
TERRY, THE NUMBER YOU ARE TRYING
TO GUESS LIES BETWEEN 1 AND 25.
(THE WORD INCLUSIVE USED HERE
MEANS THAT BOTH 1 AND 25 ARE
POSSIBLE ANSWERS.)
. XXXXXXXXXXXXXXXXXXXXXXXXX . .
!    !    !    !    !    !
0    5   10   15   20   25
GUESS #1? 16
```

In this display, the part of the line between 1 and 25 may blink on and off, or be a different color, or in some other way be prominent.

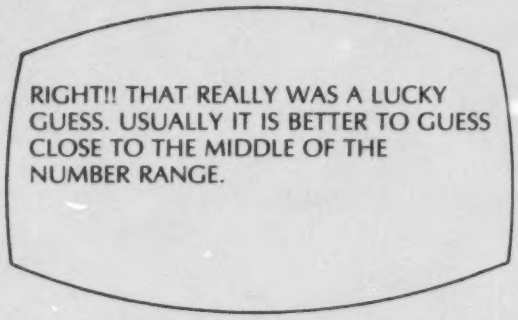
```
TERRY, YOUR GUESS #1 IS 16. THAT IS
TOO HIGH. THE NUMBER YOU ARE TRY-
ING TO GUESS LIES BETWEEN 1 AND 15
INCLUSIVE.
. XXXXXXXXXXXXXX. . . . . .
!    !    !    !    !    !
0    5   10   15   20   25
GUESS #2?
```

The machine graphically illustrates the smaller range in which the answer must lie. If a student makes a poor guess, the computer can give a suggestion. Suppose in

the above that the student's next guess is 15. The computer might respond:



Unfortunately, it may turn out by pure luck that 15 is the correct answer. The program must be "smart" enough to handle such a situation.



From the GUESS examples you can see that it is difficult to create high quality, nicely interactive, educationally sound games.

Applications

1. Play the game of GUESS with your class, with you taking the role of the computer. Use the chalkboard to record the student's guesses and your responses. The written record of the game can be used for post-game discussion and analysis. After playing the game several times with different members of the class, divide the class into teams of two. In each team designate one person to play the role of the computer and the other as the game player. The "computer" thinks of a number in the appropriate range and writes it down, and keeps a detailed written record of the game. This will help prevent later arguments. After playing a few games have the students switch roles.

   Both this activity and the one that follows are excellent mathematics education exercises, designed both to improve students' mental arithmetic skills and number sense and to give practice in writing and in organizing data by keeping a detailed written record of a game. These games are also good exercises for improving social interaction skills. Note that both participants in these activities are learning.

   Finally, you should be aware that guessing games such as these are examples of the guess and check (trial and error) technique of problem solving, which is one of the general-purpose methods of problem solving.

2. The game of GUESS can be modified so that the answer is in a different range, such as between 65 and 200. It can also be modified to allow decimal numbers (such as a one-place decimal) or fractions as answers. For example: "I am thinking of a fraction between 0 and 1. Both the numerator and the denominator are one digit numbers. What is my fraction?" These variations can be played without a computer as in 1. above.

3. If a computer is available, have your students play GUESS on it, as well as by hand. (Note: GUESS is easy enough to program so that a beginning programmer can develop a usable version. If you have access to a microcomputer, there is likely to be a version of GUESS available for this machine.) Then, have a class discussion to see which version they like best, and why.

4. If a computer is available, have a class contest to determine the best GUESS player. For example, have each student play the computer game five times. The winner is the student with the smallest total number of guesses.

5. HANGMAN is a word guessing game often played by children and now readily available in computer format. A child tries to guess the letters of an unknown word. The feedback on a correctly guessed letter is to have that letter written into its correct location(s) in the word. The feedback for an incorrect guess is an addition to a scaffold and hanging body. When the scaffold and body are complete, the guesser loses.

   Teach your students to play HANGMAN using pencil and paper. If a computer is available let them also play a computerized version. Have students write a brief report covering "how to play well." See if they discover, on their own, strategies such as first guessing frequently used vowels.

Exercises

1. Prove that in the GUESS game one can always guess the computer's number in five guesses or less by following an appropriate strategy. To do this, describe a strategy and explain why it works. Suppose that the computer's number lies in the range of 1 to 35 inclusive. Will your strategy always guess it within five guesses? Explain why or why not.

2. Read Application 1. Then discuss the relative educational merits of using a computer versus having students play the role of computer in this game. Decide which you think is best, and explain why.

3. Many children are familiar with and enjoy playing HANGMAN. It is a word guessing game, guessing one letter at a time. Feedback consists of telling whether the guessed letter is in the word, and if so, where. Try this game with some students or some adults. Analyze it as a learning/teaching aid. Computerized versions of this game are available on most microcomputers. See Application 5 above.
4. Make up a guessing game that can be played by hand, can be computerized, and has educational value. Discuss the game in sufficient detail so that a programmer could write a program for it. (If you are studying programming and have sufficient access to a computer, write the program and test it out with several people.)
5. Examine the three screens of computer output given below. Discuss the educational merits of the game. Compare and contrast with GUESS and HANGMAN.

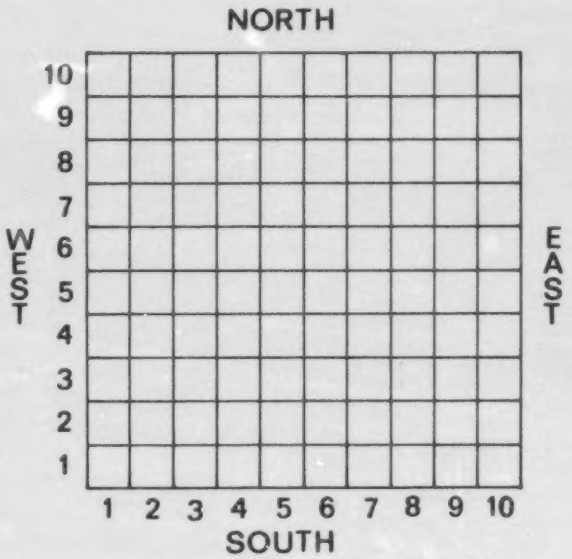
```
THE NUMBER YOU ARE TO GUESS IS AN
INTEGER IN THE RANGE 1 TO 25
INCLUSIVE.

WHAT IS YOUR GUESS?  13
```

```
THAT IS INCORRECT.
THE NUMBER YOU ARE TO GUESS IS AN
INTEGER IN THE RANGE 1 TO 25
INCLUSIVE.

WHAT IS YOUR GUESS?  8
```

```
THAT IS INCORRECT.
THE NUMBER YOU ARE TO GUESS IS AN
INTEGER IN THE RANGE 1 TO 25
INCLUSIVE.

WHAT IS YOUR GUESS?
```

HIDDEN TREASURE

A 2-dimensional variation of GUESS is a program we will call HIDDEN TREASURE. The hidden treasure is located somewhere on a 10 by 10 grid.



The game player makes a guess at the location by specifying a pair of coordinates. Mathematicians generally specify the right-left coordinate first, and the up-down coordinate second. Thus, the directions to the game players are to enter a pair of numbers, separated by a comma. The first number is the east-west coordinate of the guess, and the second number is the north-south coordinate of the guess.

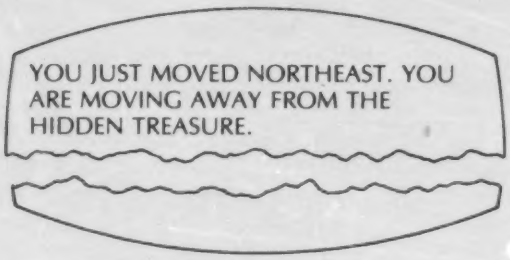
```
WHAT IS YOUR GUESS #1? 5,9
GO SOUTHWEST TO FIND THE
TREASURE.
WHAT IS YOUR GUESS #2?
```

The game player's first guess is the point (5,9). The computer analyzes this response and indicates that the hidden treasure is southwest of this guess. The game player's next guess might be the point (3,4). This is somewhat to the south and considerably to the west of the first guess.

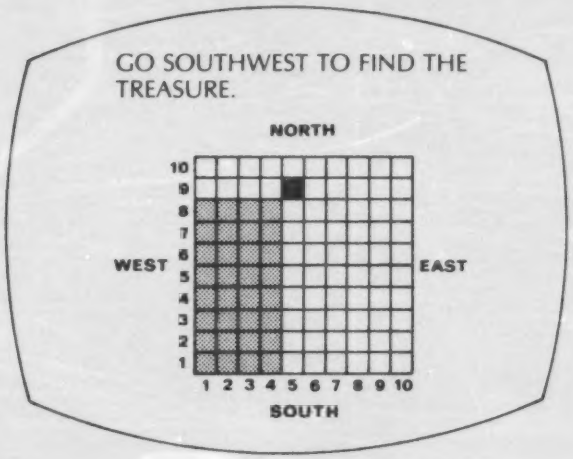
Many of our comments about GUESS also apply to HIDDEN TREASURE. This is certainly a more mathematically sophisticated game. The student is dealing both with 2-dimensional coordinates and with compass directions. Thus the opportunity for learning is greater.

As with GUESS, it is possible to program a version of

HIDDEN TREASURE so that some teaching goes on as the student uses the program. Suppose the student's initial guess is (5,9) and the computer responds with GO SOUTHWEST. If the student's next response is (7,10) the computer might respond:



The computer can be programmed to graphically display the search space. After a guess of (5,9) the machine could respond:



The most recent guess is indicated by a dark square and the shaded (or colored) region is where the treasure must lie.

A game such as HIDDEN TREASURE seems more interesting to young students than does GUESS. They will play it over and over again and take pride in finding the treasure in a small number of guesses. Many of the computerized adventure games are merely elaborate versions of HIDDEN TREASURE.

Applications

1. Play HIDDEN TREASURE with your class, with you taking the role of the computer. After students understand the game, divide them into pairs and have them play with each other. The students playing the two roles are to write their responses, so that a written record is always available for postgame analysis. Learning to keep a detailed and systematic record of one's activities is important to learning science and mathematics.
2. One can create variations on this game, and commercial games that teach related skills are available. One such commercial game is called BATTLESHIP, or perhaps WAR. It is likely that some of your students know how to play. Each side places a number of ships into a grid. Players take turns trying to locate and sink their opponent's ships. Have your students explain the game of BATTLESHIP to you and to others in the class who are not familiar with it. Note that explaining a game is a valuable learning experience. After all students know how to play, have each student write down what sorts of things they learn by playing the game. This same idea can be used with any game having educational merit. Students should be encouraged to become consciously aware of what intellectual skills they are exercising when they are playing a game.
3. A 3-dimensional version of HIDDEN TREASURE could use the compass directions plus up-down. The game then becomes BURIED TREASURE, and students can play it with each other. Carry out the activities of Application 1 using BURIED TREASURE.
4. Encourage your students to make up their own games based upon ideas of GUESS and HIDDEN TREASURE. For example, the treasure might be in any one of 37 different rooms, where one must first guess the room. The rooms may be of different sizes, and one must guess where in a room the treasure is hidden. Have each student write down his/her game.

Exercises

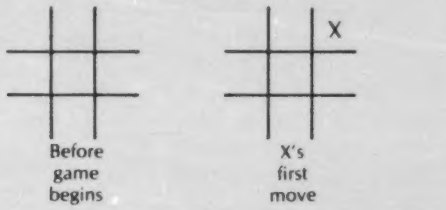
1. In HIDDEN TREASURE, the computer's response to an incorrect guess is one of eight possibilities. These are GO NORTHEAST, GO NORTHWEST, GO SOUTHEAST, GO SOUTHWEST, GO NORTH, GO SOUTH, GO EAST, GO WEST. Explain how a computer can figure out which of these responses to give. That is, imagine you are a computer programmer. You know the coordinates of the guess and the coordinates of the location of the hidden treasure. How do you use this information to figure out the computer's response?
2. Analyze HIDDEN TREASURE to determine how many guesses it takes to find the treasure, provided one uses an optimal strategy. For example, can one always find the treasure in five or less guesses? Decide what the maximum number of needed guesses is, and prove that your answer is correct.

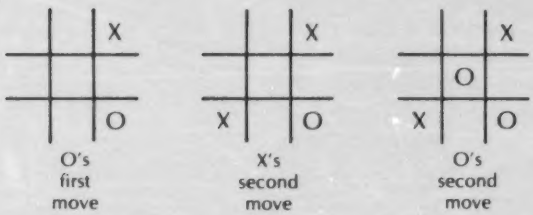
TIC-TAC-TOE

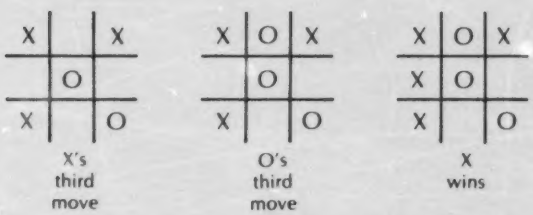
There are many common two-person competitive games such as tic-tac-toe, checkers, chess and backgammon. For any of these games a computer program

can be written so that the computer plays one or both sides. We will discuss TIC-TAC-TOE (TTT) to see how this is done, and look at the possible educational applications.

TTT is a two-player game, with players taking turns. One player is designated as X and the other as O. A turn consists of marking an unused square of a 3x3 grid with one's mark. The goal is to get three of one's marks in a file (vertical, horizontal, or diagonal).

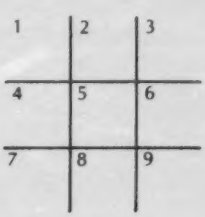






Although the game of TTT is simple, it does involve some strategy. One needs to think ahead, determining the consequences of a possible move, to play well. Of course, if one is satisfied with playing poorly, then little thinking is required; one can merely make random moves.

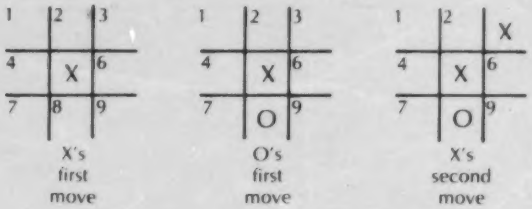
It is quite easy to program a computer to make random moves in a game of TTT. You can easily simulate this idea by hand. Prepare nine slips of paper, numbered 1 through 9. Number a TTT board's squares 1 through 9.

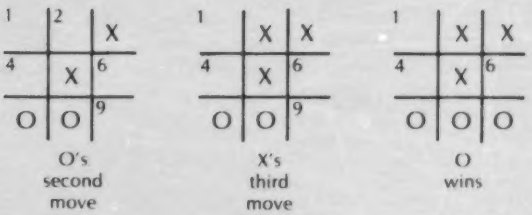


Let the "computer" go first and be X. To generate the computer's move, draw one of the slips at random. Whatever number is on the slip is the computer's move.

After you decide upon your own move, mark it on the game board and remove the slip that has that number. Thus, when it is the computer's turn to make its second move there will be only seven slips left. Each will be the number of an unused square.

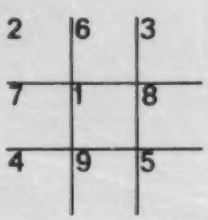
A typical game generated in this manner might appear as follows:





As you can see, X plays aimlessly and eventually loses. At X's third turn, a move to square 9 leads to a win by X, assuming a minimal level of intelligence in subsequent moves.

Usually people who write game-playing programs want the computer to play well. They are not satisfied with the aimless, random approach just illustrated. Therefore, they study and analyze the game very carefully, and then try to embody a program with their knowledge and skill. The game of TTT is sufficiently simple so that it is easy to write a procedure which leads to good play. The TTT-playing procedure given below depends upon numbering the squares of the gameboard as follows:



The procedure generates moves for X. When it is X's turn to play, decide upon a move as follows:

1. If there is a file (that is, a row, column, or diagonal) containing 2 X's and no O, play in that file. (If two such files exist, play in the file containing the lowest numbered unused space.) Otherwise:

2. If there is a file containing 2 O's and no X, play in that file. Otherwise:
3. Consider each possible remaining legal move, in increasing numerical order. For each, see if the result would be the creation of two distinct files, each containing 2 X's and no O's. If (and as soon as) such a move is found, make that move. If no such move is found then:
4. Move in the lowest numbered unused space.

This TTT procedure can be executed by a person. Alternatively, it can be programmed, and a computer can serve as the agent that interprets and carries out the move instructions. You should practice acting as the agent. Play yourself a game, with the above procedure generating X's moves and going first. You play O, and try to beat the procedure. Notice that X's first move will be in square 1, generated by step 4 of the procedure.

Suppose that you happen to respond with a move in square 6.

It will turn out that X's second move will be in square 2, again generated by step 4 of the procedure. X is threatening to win, and you block by playing in square 5.

X's next move will be to square 4, based upon step 3 of the procedure. This move guarantees that X will win. No matter how O responds, X will win on the next move; this winning move will be generated by step 1 of the procedure.

Applications

1. Have students in your class play TTT with each other by both players using random moves. Number the board squares 1 through 9 and make slips of paper numbered 1 through 9. Players take turns drawing a random number and marking the move on a 3x3 grid. Have each pair of students keep track of how many times X wins, how many times O wins, and how many times "it's the cat's game" (a draw). Collect class statistics on a chalkboard and discuss.

   In doing this activity, students gain experience in learning and following a procedure, and in record keeping. They will have fun seeing how poorly the random move generator does, and thus see that they are smarter than a random move generator. You can tie this activity in with computers by pointing out that a computer can be programmed to play a game such as TTT randomly. Just because a computer has been programmed to play a game does not mean the computer will always win, or even always play well. Computers are not inherently smart!
2. Teach your students the four step TTT playing procedure given in the TTT section. Have each student play several games against this procedure. First, do this with the procedure making the first move. Have students keep detailed records of each such game, and who wins. Collect class statistics on the results. Note that if X goes first and follows the procedure, X will never lose. Thus, if a student claims to have beaten the procedure, you will want to see the evidence. This is one reason for having each student record details of each game s/he plays.

   Next, have the student (O) go first, playing against the procedure, X. It turns out that O can win such a game. Your students will enjoy searching for a way to beat the computer procedure. Let one of your students explain how and/or why it is possible to win when O goes first.

   It is a worthwhile learning experience for students to figure out a method for keeping a record of a game. Much of science is based upon the careful collection, organization, and subsequent analysis of data. You might have several students explain their record keeping systems. After several methods have been illustrated, lead the class in a discussion of their relative merits. Is one method clearly better than others? This is a typical real world problem with lots of acceptable solutions, and no "best" answer.
3. The game of NIM is played by two people. In a typical version of NIM there is a set of 21 objects, such as matches or coins. Players take turns removing one, two, or three objects from the set. At each turn, a player must remove between one and three of the objects, the actual number being at his/her discretion. The goal in the game is to not take the last object from the set.

   Play the game on the blackboard to illustrate it to students. Then divide the class into groups of two and have them play randomly. Random numbers in the range 1-3 can be generated using numbered pieces of paper drawn from a box or hat. Have the groups and the class collectively keep track of whether the first or second player wins more often under random move conditions.

   Next, let the students play as well as they can and encourage them to develop strategies. See if a student can develop a winning strategy. Variations on this game include starting with a different number of objects, changing the goal to taking the last object rather than not taking it, and allowing a different range of objects to be removed each time, such as 1-4.

Exercises

1. In Application 2 it is asserted that if X goes first and follows the four step procedure given in the TTT section then X will never lose. Convince yourself that this is correct. Then write down a description of what you did to convince yourself that X can never lose. Your written argument should be convincing to a fellow student.

   In this exercise you are trying to prove that a particular procedure does what it is supposed to do. Computer scientists spend considerable time trying to prove that their programs are correct. Proving that a procedure "works" or proving a program is correct is quite similar to proving a theorem in mathematics.

2. In Application 2 it is asserted that if O goes first and X follows the four step procedure given in the TTT section that X can be beaten. Prove that this is correct by giving an example in which X loses. Compare the difficulty of this exercise with 1 above. It is often much easier to show something is not correct (by finding a counter example) than to prove it to be correct.

3. Suppose X plays by the four part procedure given in this section. Write a procedure to be followed by O so that every game ends in a draw. Will every game then be exactly the same? Modify your procedure, making use of random move generation in an appropriate place, so that not every game is the same.

4. Is it possible to give a procedure to be followed by the first player X, so that X never wins? Explore this problem. If you conclude that it is possible, give a procedure for X.

5. Write a computer program to play NIM (see Application 3) against a human opponent. The computer first generates a random number (like a coin toss) to determine who goes first. The computer makes its moves using a random number generator. Be sure to have your program check the legality of the human's moves. If a move must be a number in the range 1-3, then moves of 0 or 4 are illegal.

6. Obtain a computerized game of chess, checkers, or some other "hard" game. Try it out yourself and then try it with several students. How do your students react to playing against the computer? Pay particular attention to the social interaction as students team up in order to try to beat the machine. Write a report on your findings.

VIDEO GAMES

You are undoubtedly familiar with the popular home entertainment video games and their "parents" found in commercial entertainment centers. Almost all involve a display screen, moving objects, hand operated control units, sound and flashing lights. Some are small hand-held, battery powered games, while others use a regular AC power source and a television set for a display screen. In some of the games a player pits his/her skill against the machine; others are two person games, with the computer serving as a referee and score keeper. There is no doubt that these computerized games have tremendous entertainment value. What is their educational value?

An obvious, but usually overlooked educational value is that these games demonstrate computer capabilities. One can build a hand-held, battery powered, computerized machine that can play an intricate game. When mass produced, the game is relatively inexpensive (perhaps under $25 retail) and thus within the price range of many millions of people. The more expensive games illustrate the use of hand operated controllers and keyboards for computer input. They make use of quite intricate computer graphic displays, often in color. Such machines tend to be reliable and durable.

A second, and often overlooked, educational value of these process control games is the insight they provide into computer simulation. You have undoubtedly heard of computerized pilot trainers, driver trainers, spaceship control trainers, and so on. Some of the commercial entertainment center video games are merely simplified, less expensive versions of these trainers. A computer simulation of piloting a plane can be so close to the real thing that it is a substitute for a substantial part of actual flight experience. That is, the computer can simulate many aspects of the "real thing." We will discuss computer simulation in more detail later in this chapter and briefly in a later chapter; it is one of the most important applications of computers.

To learn more about the possible educational value of the process control games, let's look at one in more detail. Among the first such games, and of lasting popularity, are those which simulate certain aspects of ping pong or tennis. We will use the name PADDLE BALL for a game title. PADDLE BALL can be played against a computer or against a human opponent. One uses a hand-held controller to move a paddle on the display screen. The object is to hit a moving "ball" on the display screen. Missing the ball results in a score for your human or computer opponent.

Such games teach hand-eye coordination, two dimensional spacial visualization, and intuitive geometry. You watch as your opponent moves his/her paddle to intercept the ball. You anticipate the angle of rebound, and begin to position your paddle. Your eyes, brain and muscles must work together to solve intricate angular-rebound problems and to quickly position your paddle. Skill increases with practice.

Another possible educational value of such games is the scoring. From a student's point of view, here is a "practical" application of counting and of record keeping. A common variation of PADDLE BALL involves a player hitting a ball against a brick wall that is many bricks thick. As a brick is hit by the ball it disappears and one's score increases. The interior bricks are worth more points. One plays the game over and over again, trying to better one's previous score or the scores of other people.

We mentioned earlier that skill in playing most video games increases with practice. This is an important thing for students to learn. Each time a student encounters a new learning situation (for example, a new game or a new type of problem) the student's initial performance level is apt to be low. But with study and practice the performance level will improve. If a student goes through this cycle many times s/he will develop confidence in his/her ability to learn.

Developing confidence in one's ability to master a new situation, learning that one can learn, is one of the most important goals of education. Every student can learn to play a video game. Every student will find that his/her skill increases with practice. Every student can experience pleasure and satisfaction with this increased skill and can internalize the idea that increased levels of performance require time and practice. Thus, video games can make a significant contribution to a student's educational development.

This analysis helps explain why some students become video game addicts. The games are designed so that one gets better with practice; that is, experiences more and more success. There are no permanent records (bad grades) for not doing well. One can compete against oneself and/or against other students' high scores. If a particular game isn't much fun, one stops playing it and tries a new game. One is not forced to play a particular game. (Contrast this with school or with other aspects of life in our society!)

Applications

1. Ask your class to describe their experiences with video games, and what they like about these games. You may gain insight into the outside-of-school experiences of your students and how broad an impact video games have had. Keep in mind that traditional school must, in some sense, compete with this outside-of-school world.
2. Have students bring hand-held electronic games to school. Allow each student to play with several games. Each student is to write a brief report on which game has the most educational value (and why), and which game is the most fun (and why). Both you and your students will gain insight into how electronic games can be both fun and educational.
3. If a computer is available, have your students play one or more games. Have a class discussion in which students state which games they like best, and why. How do the computer games compare in popularity with the hand-held electronic games?

Exercises

1. Play several different video games. (This may involve a trip to your local computer store, department store or commercial entertainment center.) Describe each and discuss its possible educational value. How could each game be modified to increase its educational value?
2. Design (in your mind) a video game with entertainment value comparable to commercially available video games, but educational value comparable to usual classroom instruction. Describe your game in considerable detail, pointing out its educational and entertainment features. If you are a good programmer, program your game.
3. In the past section we discussed ideas such as learning that one can learn and learning that skill increases with practice over time. Give some examples of how the traditional school curriculum deals with these ideas. Suggest some changes in this curriculum that might lead to increased student insight into learning about learning.
4. Dungeons and Dragons is a widely played game involving imagination, record keeping, and group interaction. Although Dungeons and Dragons was originally a paper and pencil game, computerized variations are now widely used. Students spend literally hundreds of hours mastering and playing the game. Investigate this phenomenon. What do students learn by playing Dungeons and Dragons? Is what they learn considered relevant to the school curriculum? (Some schools teach D&D as part of their curriculum and encourage students to participate in D&D groups.)

SIMULATIONS (RULER)

Many of the ideas discussed so far in this chapter can be combined to construct very intricate simulation games. Since simulation games are often designed especially for education, and since many educators frown upon allowing students to play games, we will use the shorter name simulation.

The computer simulation RULER allows the game player to be the ruler of a country. Imagine yourself as ruler of an ancient country, making decisions as to how to allocate resources. How much grain should be planted, and how much used to feed your people? How many people should be farmers, how many should be soldiers, and how many should build monuments to appease the gods or to increase the glory of their ruler? How does your country cope with poor or abundant harvests, increasing or decreasing population, enemy invasions or natural disasters? Good decisions, plus some good luck, can lead to an increased popularity and long reign for you. A poor showing on your part leads to a revolution.

A computer simulation such as RULER is exciting. The simulation may be historically accurate, giving good insight into life in ancient times. Making correct decisions may require a careful analysis of resources available, and may require taking into consideration the probabilities of various events occurring. Good math skills are needed to play well. Thus RULER is both fun and educational.

Computer simulations can add a new dimension to education. They can put a student into an environment that would be difficult or impossible to bring into a classroom. They can create problem situations of an authenticity and relevance not readily duplicated by other educational aids. This is particularly important since most problems presented to students in their school work are both contrived and rather silly. The typical textbook problem is not representative of the real world nor is it inherently interesting to students. Computer simulations can be rich in problem-solving requirements and opportunities; substantial learning can occur in such a highly motivating environment.

There are many simulations commercially available, and more being developed all the time. But the development of an educationally sound simulation is difficult and quite time consuming. It requires a good knowledge of the situation to be simulated, learning theory, and computers. As with other computer assisted learning materials, a team approach is quite commonly used. In any event, a single high quality computer simulation can easily require 500 to 1000 hours or more of development time on the part of highly qualified individuals. Thus, it is not surprising that most of the best computer simulations are aimed strictly at the entertainment market, where potential sales are currently much larger than in the education market. Moreover, most currently available educational simulations were written for secondary school and college students. Again, this is because at the time of their writing the potential audience of elementary and middle school student users was very small. But now that computers are becoming available for use in the lower grades there will be an increasing number of simulations written for younger students.

A simulation need not depict a real world situation. One can simulate imaginary situations such as wars in outer space, looking for hidden treasures in a land of caves and dragons, or adventures in imaginary countries. There are many excellent examples of this type of software commercially available. This type of computerized entertainment is more than a match for television with a growing collection of children (and adults) who have easy access to computers. They will play for hour after hour, actively involved in trying to figure out details of the game and how to improve upon their previous performances.

Exercises

1. Play some computerized simulation-games and/or observe some students and adults playing such games. Discuss what you learn, paying special attention to the potential educational merit of the simulations.
2. Design (in your mind) a simulation that would have substantial educational value and would also be fun to use. Describe it in some detail. Discuss its educational value relative to more conventional means for teaching the same materials.

CHAPTER FIVE

AUTOMATED SYMBOL MANIPULATION

A computer is a machine designed for the input, storage, manipulation and output of symbols. The symbol set used inside a computer consists of letters, digits and punctuation. A computer can function automatically, rapidly and accurately following detailed step by step directions stored in its memory. In other words, a computer is a high-speed, automated symbol manipulator.

The human brain, often aided by paper and pencil, is also a symbol manipulator. It is more versatile than a computer, but not as fast or accurate. Thus, computerized symbol manipulation can be a valuable aid to human intellectual activity.

Historical Sketch

Human beings are blessed with considerable intelligence and sophisticated language skills. A sound (spoken word) can be used to stand for an object. A sequence of sounds forms a sentence, allowing complex ideas to be communicated. With spoken communication, information is preserved from one generation to the next, and an accumulation of knowledge is possible. Early human beings developed this skill, perhaps hundreds of thousands of years ago.

But the human memory has limitations, and accurate memorization of a large amount of information is hard work. Thus, it is not surprising that people began to develop memory aids, written symbols, to supplement pure memorization. Charcoal marks on a cave wall could stand for the passing days, aiding in the prediction of when the moon would again be full. Pictures of successful hunts could serve to remind people of great events of the past.

Eventually the time in human history was right for the development of agriculture—raising crops and animals. This increased the need to measure time, to count, and to measure land. Time measurement was important so that crops would be planted and harvested when the weather was apt to be appropriate. Counting was useful in keeping track of quantities of animals and of stores of farm produce. Geometry (earth measurement) was necessary to keep track of the location of one's lands, especially in areas that were flooded each year.

The development of agriculture also allowed for a larger leisure class—people who did not have to spend all of their time at physical labor in order to have enough to eat. There was more time for thinking, learning, accumulating knowledge. More sophisticated written symbol systems were developed; reading and writing were born. Often the accumulated knowledge seemed like magic to the uneducated, adding to the prestige and power of the priests, witch doctors, and scholars who were able to use it. Very early in human history it was realized that knowledge is power.

Almost all humans easily master the spoken language of their community, often developing considerable skill by age two or three. Counting and simple mental arithmetic are also learned without much benefit of formal instruction. But the use of written symbols to stand for words and numbers is not so simple. Years of instruction and practice are required to develop good skills in these areas which are part of the "basics" of a modern education.

Moreover, as societies became more complex, the problems to be represented and solved using written symbols became more complex. Thus human ingenuity began to look for aids to solving the problems. We shall briefly discuss three types of aids. Think carefully about these aids, since they provide useful insight into our educational system.

First, consider the symbols themselves. You are familiar with Roman numerals.

I II III IV V VI VII VIII IX X XI XII XIII

The Roman numeral system is useful for dealing with relatively small, positive integers. But it is severely limited. Can you count to a million using these symbols? Can you represent and deal with zero, negative

numbers, fractions? The Hindu-Arabic numeral system we now use is far superior to the Roman numeral system. It took thousands of years for mathematicians to develop the number system we now teach to young children.

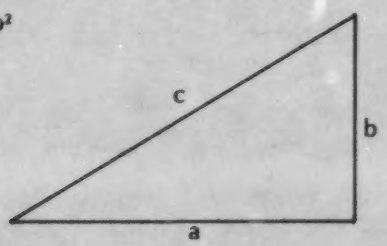
One might make similar comments about an alphabet-based writing system versus a picture-based system. To your author, at least, the learning of a 26 character alphabet seems simpler than learning the thousands of characters in a language such as Chinese or Japanese. However, young Chinese and Japanese students master these languages, so it must not be too difficult.

But think of how much easier spelling would be if every word was spelled exactly the way it sounds. Think of how much easier grammar would be if there were no irregular verbs. Our written language system is useful, but leaves a lot to be desired!

The field we now call mathematics is representative of a second type of progress in dealing with hard problems. Consider, for example, one of the difficult problems of land measurement. Right angles must be precisely constructed. One way to solve this is to tie equally spaced knots in a rope, and then make a three-four-five triangle. The result is a right angle.

A generalization of this idea is the Pythagorean Theorem. If a, b, and c are lengths of sides of a right triangle (c is the longest side) then

$c^2 = a^2 + b^2$



This fact is very useful in surveying and navigation.

Mathematics gives us symbology to represent many complex problems and ideas. It gives us algorithms to add, subtract, multiply and divide whole numbers, fractions and decimal fractions. Mathematics is a tool essential to engineering and science. The development of a mathematical idea may be enough to immortalize a person (Euclid's geometry; Pythagorean Theorem). But the crucial point is that people with only ordinary ability in math can build upon the work of the mathematical geniuses. High school students of today routinely solve math problems that stretched the capabilities of the greatest mathematicians of Euclid's time.

A third wave of aid to problem solving is machines. The abacus was developed about 5,000 years ago and eventually became a common aid to calculation. It has proven to be such a valuable aid that its use continues even today. Many elementary school teachers find that an abacus or bead frame is helpful in teaching addition, subtraction, place value, carrying and borrowing. Storekeepers in some countries still use an abacus as an everyday tool.

It is interesting to examine computation in Western Europe in the early 1600s. The abacus was a standard tool. But the Hindu-Arabic numeration system was well entrenched, along with paper and pencil algorithms for addition, subtraction, multiplication, and division similar to those we currently use. A philosophical struggle arose between those favoring use and teaching of the abacus versus those favoring the teaching of paper and pencil algorithms for arithmetic. In Western Europe the latter group won out, and use of the abacus nearly disappeared. In Eastern Europe and in Asia use of the abacus continued.

But the paper and pencil algorithms were difficult to learn, requiring considerable memorization of number facts (the "times" table, for example) and considerable practice. Thus, while many people learned simple addition and subtraction, multiplication and division were advanced topics offered in higher education. Many quite educated people of that time period could not solve multiplication and division problems of the sort that students currently in our grades 5-7 are expected to handle. (Perhaps we should not become so discouraged when young students are not able to master these difficult algorithms.)

Not surprisingly, then, the 1600s saw the development of mechanical digital calculators that could perform the four basic arithmetic calculations. Initially such machines were hand built, one-of-a-kind, expensive and unreliable. But the technology of the time was adequate, and eventually good quality calculators became available in government and business. Such machines were sufficiently expensive that their availability in, and impact upon, general education was nil.

One of the early calculators which could add and subtract was built by Blaise Pascal, who also made major contributions to the mathematical knowledge of his time. A modern programming language, often used in college courses for computer science majors, has been named Pascal in his honor. Pascal is often credited with being the first person to build a calculator. This is certainly not correct, since there is solid historical evidence of Wilhelm Schickard having built a four function calculator in 1623, the year in which Pascal was born.

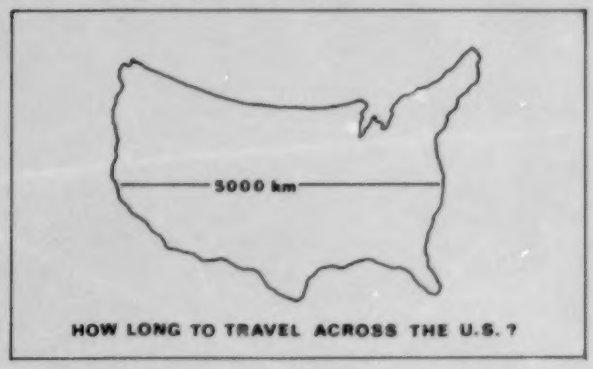
In the 1820s and 1830s, Charles Babbage, an Englishman, began trying to build a general-purpose computer. Babbage envisioned an automated calculator, able to carry out a sequence of calculations and store the intermediate results. Although the technology of his time was not adequate to the task, Babbage developed many of the principals of modern computers. Ada Lovelace, a daughter of the poet Byron, developed ideas of computer programming for the machine Babbage was trying to build. She is acknowledged as the world's first computer programmer. A programming language that has been developed for and adopted by the United State military services is called ADA in her honor.

About a hundred years after Babbage's work, in the late 1930s, people again began to develop general-purpose digital computers. An electromechanical machine, called the Mark I, was developed by Howard Aiken at Harvard University and became operational in 1944. This machine did what Babbage had envisioned. However, it was quite slow, requiring several seconds to complete a single arithmetic operation.

The first general purpose electronic computer built in the United States, called the ENIAC, was developed by John Mauchly and J. Presper Eckert at the University of Pennsylvania. The ENIAC became operational in December, 1945. It used vacuum tubes for switching circuitry and was several hundred times as fast as the electromechanical machine developed by Aiken.

Many of the basic ideas for an electronic digital computer were developed by Joseph Atanasoff at Iowa State College (now Iowa State University) during the late 1930s and early 1940s. Significant progress also occurred in England and Germany during this time. As with many inventions, nearly parallel development went on in several distinct locations.

Both the Mark I and the ENIAC were designed to automatically, rapidly and accurately carry out a sequence of calculations. The ENIAC was approximately a thousand times as fast as a person using a desk calculator. A change of this order of magnitude is revolutionary! Consider the speed of a person walking, say 5.5 km/hr; the speed of a car, say 55 km/hr; the speed of a propeller driven airplane, say 550 km/hr; the speed of a supersonic jet plane, say 2,200 km/hr. This total range of speeds represents a factor of 400 from the slowest to the fastest. The first electronic digital computer represented a much larger factor of change, and it was only the beginning. The fastest computers of today can multiply or divide a billion times as fast as a person using pencil and paper.



HOW LONG TO TRAVEL ACROSS THE U.S.?

After completing ENIAC, Eckert and Mauchly continued to develop computers. They developed the 1951 UNIVAC I, which was the first commercially available general purpose computer. This was a vacuum tube based machine, as were all of the early computers. But the transistor was invented in 1947 and the idea of using tiny donut-shaped electromagnets for primary memory was developed at nearly the same time. It took about a dozen years to get these devices into mass production and widespread use. By 1960, good quality, reliable machines, employing transistorized circuitry and magnetic core primary memory, were on the market. Thousands of these machines were sold, and the age of computers was well underway.

In tracing the history of computers it is easy to become mesmerized by hardware progress. But software progress has also been substantial. The ENIAC was programmed by connecting wires between memory locations, much like on an old-fashioned telephone switchboard. John von Neumann and others suggested the idea of storing programs in the primary memory. Out of this came the idea of machine language, the language a particular machine constructed to be able to "understand."

Machine language instructions differ between different brands of computers. Moreover, it is quite easy to make a small error while doing machine language programming. For example, one may code an instruction as 80903521793 when intending to write 80903251793. The smallest error of this sort generally results in a program that doesn't work right.

One simple aid, assembly language, was immediately developed. Abbreviations such as ADD, SUB, MUL and DIV were substituted for numerical codes representing addition, subtraction, and so on. Memory locations could be given names such as PRICE, HOURS, DATE and ACCOUNT. A computer program, called an assembler, could translate assembly language programs into machine language programs. Assembly languages and assemblers increased programmer productivity, decreased programming errors, and made it easier to learn how to program.

The idea of developing software to make it easier to learn to program and easier to program has been a driving force in the computer field. In the early 1950s it took a year or more for a person to learn an assembly language and develop a reasonable level of skill in its use. Most people were not willing to devote this much time and effort to learning how to use a computer. Better languages, better human-machine interfaces, were needed.

FORTRAN, developed during 1954-1957, is an excellent example of a better language. It was designed for scientists and uses the mathematical notation familiar to all scientists. A physicist or engineer can develop a useful level of FORTRAN programming skill in a modest number of hours of instruction.

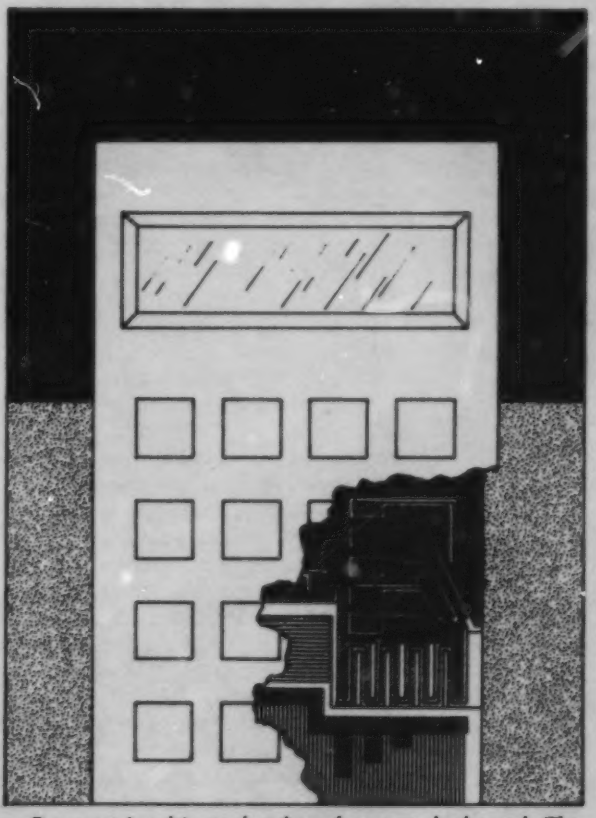
FORTRAN was the first widely adopted "higher level" language—that is, a language more sophisticated than assembly language. It was followed by COBAL, developed for business data processing; ALGOL, another science-oriented language; and a variety of other languages. FORTRAN, COBOL, and ALGOL have been widely adopted throughout the world and are still in widespread use. All of these languages have the advantage of being machine independent, or nearly so. A FORTRAN program written to run on one brand of computer is apt to run correctly on a wide variety of other computers. A programmer learns one language, and can use that language on a variety of machines.

As computers became cheaper and more readily available, it became evident that a person did not need to be a graduate level scientist or engineer to use one. The language BASIC was developed at Dartmouth College in the early 1960s for use by undergraduate students in all disciplines. Its use quickly spread to high schools and to junior high schools. With proper textual materials, teacher help and computer access, even elementary school students can learn a language such as BASIC.

You should be aware, of course, that BASIC was developed for college students. Languages more suited to the needs of elementary school students have been developed, and new ones will be developed in the future. The language Logo, developed by Seymour Papert of Massachusetts Institute of Technology, is gradually gaining acceptance; it is specifically designed for elementary school students, and is now available on several different brands of microcomputers. Research on the use of Logo with grade school students suggests that it is an excellent aid to learning in a variety of disciplines such as math, science, language arts, art and music, as well as in learning about computers.

Progress in transistor technology during the 1950s and 1960s led to mass production of transistors, then to integrated circuits. Initially an integrated circuit contained the equivalent of a few dozen transistors and other electronic components, but cost about the same to produce as a single transistor. The price of computers went down—as did the price of transistor radios and other electronic equipment.

Eventually it became possible to manufacture a single circuit containing the equivalent of thousands of transistors and other electronic components. This was called a large scale integrated circuit (LSI). This progress led to minicomputers (smaller, less expensive computers) in the latter part of the 1960s, and to hand held calculators in the 1970s. Since about 1975, hand held calculators have been cheap enough to be a common household item, and to be available in elementary and middle schools.



Progress in chip technology has not slackened. The mid 1970s saw the introduction of the first microcomputer, and by 1980, hundreds of thousands of these computers had been sold. The first hand held computer, programmable in BASIC, became commercially available in 1980. It is the size of a hand held calculator, but contains a full alphabetic keyboard

in addition to the numeric symbols. By early 1981, several easily portable microcomputer systems were commercially available. The last chapter of this book looks at the future. Continued rapid progress of both hardware and software can be expected.

Applications

1. Have students study various numeration systems and write brief reports on how to do arithmetic in these systems. For example, how does one add numbers represented using Roman numerals? How does one multiply numbers represented using Roman numerals?

   A numeration system, such as the Hindu-Arabic system now commonly used throughout the world, is not "God given." Rather, it is developed by people over a period of many years. The current system is widely used because it is better than earlier systems. As students explore early numeration systems, they can compare them with the Hindu-Arabic system to see advantages and disadvantages.

   Some students enjoy developing their own sets of numerical symbols and ways of representing numbers. You might have a contest to see who can develop the most interesting, unique or useful numeration system.

2. How do we build upon the work of others? Discuss this general concept with your students, and give a few examples such as via our language and numeration system, our writings (textbooks, poetry, etc.), music and art, machines and prepackaged cake mixes. Then have each student do a brief report on how the work of people in the past affects them now.

3. The transistor is one of the great inventions of science. What can a transistor do, and where are transistors used? Develop a science unit focusing on these ideas. Students can collect pictures, bring in equipment containing transistors, write reports and in other ways increase their awareness of the widespread use of transistors.

   When transistors first became commercially available in the early 1950s, a single transistor cost more than the vacuum tube it was designed to replace. But the transistor was more reliable and rugged and used much less electrical power. Now, a single large scale integrated circuit may contain the equivalent of 50,000 transistors, and yet cost just a few dollars. (In early 1981, Hewlett-Packard Corporation announced that it had developed a very large scale integrated circuit containing 450,000 transistors.) The heart of an inexpensive hand held calculator is a chip (a large scale integrated circuit) costing perhaps one dollar. Such chips are now used in radios, television sets, automobiles and in thousands of other devices. It is no wonder that the inventors of the transistor were awarded a Nobel prize!

Exercises

1. Think back over your formal education. Have you received explicit instruction on the idea of building upon the work of others to solve problems? Write a short paper discussing the nature and extent of how this idea was explicitly or implicitly built into your formal education, and discuss the educational implications of how our current curriculum treats this topic.

2. Select one of the four basic arithmetic operations and give two different paper and pencil algorithms for doing the calculation. Compare and contrast the two algorithms. Make sure you discuss such things as ease of learning, ease of use, speed, ease of remembering, and value as a foundation for future learning. In what sense is using these algorithms merely symbol manipulation; that is, how much thinking and understanding is required to use the algorithms?

3. It is possible to estimate the time of day by examining the sun, use of an analog watch, use of a digital watch or by dialing the "time" information service on a telephone.

   a. Discuss the difference between being able to determine the time of day and understanding or knowing the meaning of time.
   b. Compare and contrast various methods of telling time (see 2. above).

4. A calculator can be thought of as a machine that can manipulate numerical symbols. When hand held calculators became cheap and reliable, acceptance by adults for use by adults was widespread. Most homes now contain a calculator, and sales in the United States probably exceed 30 million calculators per year. But similar rapid acceptance into elementary and middle schools has not occurred. Give your opinion as to why. Predict what you think will happen with calculator usage in schools over the next ten years, and give arguments to support your prediction.

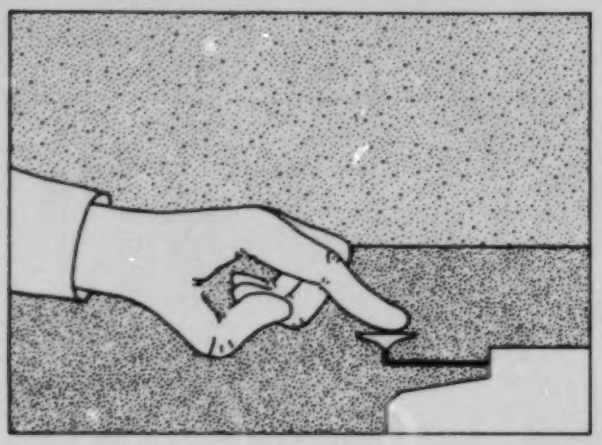
5. Musical notation provides a good example of the development of a language to aid communication in a particular field. Study the history of the development of musical notation. Discuss the quality of our current system. Is it adequately powerful, easy to use, easy to learn, easy to remember? Can you suggest improvements in this system?

Symbols

A computer is a machine designed to input, store, manipulate, and output symbols. At the most elemental level almost all computers are designed to work with just two symbols, which we designate by 0 and 1. Within a machine these symbols may be represented by a switch open or closed, current flowing or not flowing, a positive or negative voltage, and so on.

You are probably familiar with Morse code, which makes use of dots and dashes, or short and long tones.

. – a	– . . d
– . . . b	. e
– . – . c	Etc.



Using just the dot, dash, and spaces between groups of dots and dashes, all of the letters, digits and punctuation marks of standard written English can be represented. From this it should be evident that a computer that can work with just the symbols 0 and 1 is in no sense limited; one can code other larger symbol sets using 0's and 1's.

There are various binary coding schemes, methods of using the symbols 0 and 1, to represent elements of a larger symbol set. One method used to code numbers is via positional notation. Thus, one can count in binary in a manner similar to counting in base 10 notation. See Table 1.

Binary Numbers	Base 10 Numbers
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
Etc.	Etc.

Table 1. Counting in Binary.

Character	EBCDIC	ASCII
0	1111 0000	0011 0000
1	1111 0001	0011 0001
2	1111 0010	0011 0010
3	1111 0011	0011 0011
4	1111 0100	0011 0100
5	1111 0101	0011 0101
6	1111 0110	0011 0110
7	1111 0111	0011 0111
8	1111 1000	0011 1000
9	1111 1001	0011 1001
A	1100 0001	0100 0001
B	1100 0010	0100 0010
C	1100 0011	0100 0011
D	1100 0100	0100 0100
E	1100 0101	0100 0101
F	1100 0110	0100 0110
G	1100 0111	0100 0111
H	1100 1000	0100 1000
I	1100 1001	0100 1001
J	1101 0001	0100 1010
K	1101 0010	0100 1011
L	1101 0011	0100 1100
M	1101 0100	0100 1101
N	1101 0101	0100 1110
O	1101 0110	0100 1111
P	1101 0111	0101 0000
Q	1101 1000	0101 0001
R	1101 1001	0101 0010
S	1110 0010	0101 0011
T	1110 0011	0101 0100
U	1110 0100	0101 0101
V	1110 0101	0101 0110
W	1110 0110	0101 0111
X	1110 0111	0101 1000
Y	1110 1000	0101 1001
Z	1110 1001	0101 1010
blank	0100 0000	0010 0000
-	0111 1110	0011 1101
+	0100 1110	0010 1011
–	0110 0000	0010 1101
*	0101 1100	0010 1010
/	0110 0001	0010 1111
(	0100 1101	0010 1000
)	0101 1101	0010 1001
,	0110 1011	0010 1100
.	0100 1011	0010 1110
$	0101 1011	0010 0100
'	0111 1101	0010 0111
	0100 1100	0011 1100
	0110 1110	0011 1110
:	0111 1101	0011 1010
;	0101 1110	0011 1011
"	0111 1010	0010 0010

Table 2. ACSII and EBCDIC Codes.

Table 2 contains two widely used coding schemes: the seven bit ASCII code and the eight bit EBCDIC code. ASCII stands for American Standard Code for Information Interchange. With a seven bit code, one can represent $2^7 = 128$ different characters. EBCDIC stands for Extended Binary Coded Decimal Interchange Code. With an eight bit code, one can represent $2^8 = 256$ different characters.

The typical computer uses a binary positional notation to represent numbers and to do arithmetic. It also uses a code such as EBCDIC or ASCII to store characters, especially for non-arithmetic manipulation. Other coding schemes may also be used, so a computer system may have several ways of coding the same information.

You may think that we are about to launch into a discussion of binary arithmetic, conversion of numbers between base 10 and base 2, arithmetic in other bases, and so on, but we aren't! Most computer users (and even many computer professionals) do not need to understand these topics in order to make effective use of computers. Modern computer systems are designed to automatically handle conversions between base 2 and base 10. They automatically convert between our English alphabet symbols and the ASCII or EDCDIC binary codes. As far as the casual computer user is concerned, the computer works with base 10 numbers and with the English alphabet.

Applications

Many children enjoy working with codes, writing and decoding secret messages. Activities such as those given below help increase insight into the use of symbols to represent information.

1. Make a copy of the Morse code available to students. Put a coded message on the chalk board for students to decode. Try posting a "message of the day" each day for a week or so. Make it of interest to your students and instruct them not to tell other students what it says.

2. A very simple numerical coding scheme for the English alphabet is to use two digit numbers.

11 A	18 H	25 O	32 V	39 ,
12 B	19 I	26 P	33 W	40 ?
13 C	20 J	27 Q	34 X	41 !
14 D	21 K	28 R	35 Y	42 Etc.
15 E	22 L	29 S	36 Z	43 Etc.
16 F	23 M	30 T	37 .	44 Etc.
17 G	24 N	31 U	38 blank space	

(In our table, 42, 43, 44 are intended to indicate you can add more symbols.) With this coding scheme the message SEE THE BIG CAT! becomes:

291515383018153812191738131130413838

This might be broken into groups to make it easier to decode and to help prevent errors. In three character code groups the message is:

291515 383018 153812 191738 131130 413838

As with application 1, show your students the coding scheme and give them a message to decode. Have each student make up a message, code it, and give it to another student to decode.

3. Students can make up their own coding schemes, and generally enjoy doing so. Have each student make up a code and write a message. See if students can decode these messages without knowledge of the coding schemes.

Exercises

1. Represent the base 10 number 15
   a. Using ASCII
   b. Using EBCDIC
   c. Using Roman numerals
   d. Using base 2 positional notation

2. Examine a Hollerith card (often called an IBM card) that has been punched with alphabetic and numeric characters. Explain the coding scheme—the pattern of holes—used.

3. There are exactly two single bit symbols 0 and 1. If grouped into two bit symbols, there are four symbols 00, 01, 10, 11. If grouped into three bit symbols there will be eight different symbols. List them. How many four bit symbols are possible? How many five bit symbols? Suppose a binary code is to contain all upper case and lower case letters and the other symbols on a standard typewriter. Will a six bit code suffice? Why?

4. Consider coding the digits 0, 1 . . . 9 by letters A, B, . . . J. That is, 0 is represented by A, 1 by B, and so on. Write out the table of one digit multiplication facts using this code. How long do you think it would take you to memorize this table? (To a certain extent, your memorizing this table is much like a young student memorizing the multiplication facts. Rote memorization of meaningless material is difficult.)

5. One way to convert a decimal number such as 8,972,641 into base 2 positional notation would be to extend Table 1. But this would take a very long time. Another way would be to use a conversion algorithm. Suppose that you don't know any such algorithm. Do you think you could go to a library and look one up? Try it. This is an interesting information retrieval problem. Our mathematics education system is not good at teaching students how to use a library to find mathematical information. Eventually much information storage and retrieval will be computerized. Still, it will be necessary to

help students learn what information is stored and how to retrieve it.

Automated Symbol Manipulation

We began this chapter by defining a computer as an "automated symbol manipulator." Think about the steps you go through in the pencil and paper multiplication of two numbers such as 759 x 37.

```
  759
x 37
-----
 5313
2277
-----
28083
```

Very likely you can perform such symbol manipulation rapidly, accurately, with little thought. That is, you have been trained to have a computer-like capability in arithmetic calculation. A hand held calculator has the same capability, but is likely both more rapid and accurate than you. An electronic digital computer also has the same capability, but may be thousands or even millions of times as fast as a calculator.

Next consider what happens when you write. Your mind conceives of an idea and formulates it into words. Automatically, with little thought, you spell out the words using the English alphabet. You leave spaces between words, insert punctuation and form sentences into paragraphs. Writing is a form of symbol manipulation. Certain aspects of writing such as spelling of individual words, indenting the start of a paragraph, capitalizing the first word of a sentence, leaving a blank space between words and ending a sentence with a punctuation mark may be quite automatic. But conceiving and formulating ideas and organizing these ideas as sentences and paragraphs require careful thought.

This is an important idea. Certain parts of math problem solving are merely symbol manipulation that can be automatically and rapidly performed by a machine. Other parts require careful thinking; indeed, that is the essence of math problem solving. Similarly, certain parts of writing are merely symbol manipulations that can be done with little thinking, and that can be greatly aided by a typewriter or a computer. But the essence of writing is the conception and formulation of ideas—the high level thinking. This cannot be done by a machine.

Much of the accumulated knowledge of the human race is represented as symbols written in books and journals. Eventually most of this will be stored in computerized data banks and will be readily available to people who use computers as an aid to problem solving. Certain aspects of problem solving are "merely" automated symbol manipulation. The idea is that people who use both their brains and machines are much more apt to solve difficult problems than people who are restricted to just one of these tools.

Exercises

1. Check with five adults to see the last time they did a paper and pencil long division of two digit or larger numbers (other than for teaching students how to do it). Find out from each person his/her opinion on allowing students to use calculators on this problem, rather than mastering a paper and pencil algorithm. For additional information, read the article by Grayson Wheatley in the December 1980 issue of *The Arithmetic Teacher*. Then write a paper on your findings and conclusions about the importance of having students master a paper and pencil symbol manipulation algorithm for long division of multidigit numbers.

2. The library of a medium sized university may contain a million books and bound journals, and increase its holdings by 3% a year. Estimate how many years it would take you, reading eight hours per day, to read one year's acquisitions of such a library.

   An alternative to memorizing the contents of a library, or even reading all of it, is to know what is available, how to find it and how to read it when needed. Compare and contrast this idea with the long division and calculator ideas of Exercise 1.

3. In this section we discussed how certain aspects of writing and of math problem solving are merely symbol manipulation, usually done automatically with little thinking. Other aspects are highly creative, requiring high level thinking. Develop this analogy for music performance/creation, and then for one other area of your choice.

4. The astute reader may object to the automated symbol manipulation ideas of this chapter as being too narrow. How does this chapter relate to computer graphics, to drawing color pictures, graphs and charts?

   The output mechanism for a computer can be a color television display. The screen consists of many thousands of small spots or dots. The spots can be of a specified color and intensity. Thus, a picture or graph can be stored in a computer memory as a collection of numbers representing colors and intensities.

   Are there aspects of art that could be considered symbol manipulation? What aspects of art education and/or art might lend themselves to automated symbol manipulation?

5. A very useful diagnostic and research tool is having a child verbalize as s/he carries out the steps in solving a problem. The child verbalizes thoughts and explains why each step is being done. Try this with several children doing a multidigit multiplication or division calculation. To what extent does the verbalization indicate rote memory of an algorithm, and to what extent is knowledge of why the algorithm works being displayed?

CHAPTER SIX

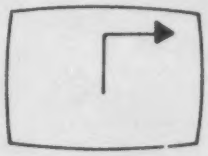
PROBLEM SOLVING IN A COMPUTER ENVIRONMENT

Remember back in Chapter 1 when you observed Terry, a third grade student, using a computer for drill on number facts? After completing the drill, Terry continues to type. You ask, "What're you doing now?" Terry replies, "I'm loading Logo. Let me show you what it can do. It's neat!"

Soon the display screen is clear except for a small triangle near the center.

Terry types FORWARD 70 and the triangle moves toward the top of the screen, leaving a line in its wake.

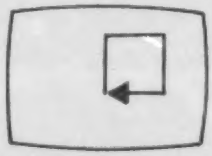
Terry types RIGHT 90 and the small triangle rotates 90 degrees to the right. The next command, FORWARD 70, draws a horizontal line.



You continue to observe as Terry types

```
RIGHT 90
FORWARD 70
RIGHT 90
FORWARD 70
```

Terry proudly points out the resulting square.



Upon further questioning you learn that Terry figured out how to draw a square "a long time ago" and is now engaged in trying to figure out how to draw a house. From the determined and eager look on Terry's face you leave, confident that Terry will probably succeed—and not only in drawing the house.

What Is Computer Programming?

Terry was writing instructions in a form that can be interpreted and carried out by a computer. Thus, Terry was programming a computer.

Computer programming covers a wide range of activity, from what Terry was doing up to and including the development and implementation of languages such as Logo, BASIC and ADA. There is obviously considerable difference between a young child exploring the rudiments of Logo and a professional programmer designing a computer system to be used by thousands of people throughout the world. It is much like a child laboriously writing his/her first words and simple sentences versus a successful author writing a novel.

Certain ideas are inherent to computer programming. They must be faced by the rank amateur and the experienced professional. Among them are:

1. A computer hardware environment—dealing with the physical machinery.
2. A computer software environment—dealing with a programming language that can be "understood" by the computer hardware.
3. Problem solving—figuring out how to solve a particular type of problem using the hardware and software one has available.

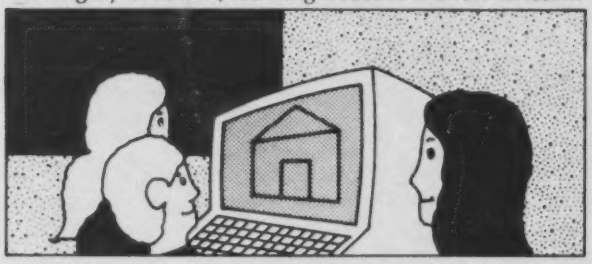
Our children are growing up in an environment which includes non-computer hardware such as the electric light, stove, refrigerator, television set, radio, stereo, car, airplane, hand-held calculator and telephone. Children easily adjust to all of these "natural" hardware parts of their environment. It can also be this way with computers. Children who grow up in a computer-rich environment learn what they need to know about the hardware. Very little formal instruction is needed, and often this can come from the child's peers.

A computer software environment includes programming languages. These languages are designed for precise communication between people and

machines. Although a computer programming language may be English-like in its general nature, its effective use requires a high level of precision. Most people require considerable training and experience to develop the necessary level of skill to write programs of substantial complexity.

The difficulty of learning a programming language varies with the language. We know that the human brain has an innate capacity to learn a natural language such as English, Spanish or Russian. We conclude that natural languages agree with learning skills inherent to the human brain. A child growing up in a language-rich environment learns to speak a language very skillfully without benefit of formal schooling.

The Logo programming language, developed by Seymour Papert, was designed with the ideas of natural language acquisition in mind. If a child is placed in a computer-rich Logo environment, surrounded by others who are using the facility, and given adequate time and encouragement, the child will learn substantial parts of the Logo language. This is learning by doing, learning by imitation, learning because it is *fun* to learn.



Such learning is in sharp contrast to the way most adults learn a programming language or a natural language. An English-speaking adult can learn Spanish or Russian, but it is usually quite hard work. In many ways, adult natural language instruction is similar to adult computer programming language instruction. Adults emerge from the experience with the idea that learning a language is hard work. If they have been learning a programming language, their conclusion is apt to be that only adults can accomplish such a difficult task.

We know this is incorrect. Children, even at a preschool level, can learn to program. Of course, we don't expect such children to write professional level programs that will be widely sold and used. But we don't expect a first grader to write a great novel or poetry of lasting value. A child can learn to write programs that solve problems from the child's world. This can be a vehicle for learning about a wide variety of topics, including computers. Terry is learning both geometry and programming by drawing a house using Logo.

Applications

The applications given here assume that you and your students have access to computer facilities. It is likely that the amount of access per student is severely limited, so that you cannot immerse your students in a computer-rich environment. Also, currently available computer hardware and software is by no means child-proof. Some adult supervision and assistance is therefore necessary.

1. Select a computer program suited to the interests and level of your students and load it into the computer system. Show one student how to use the program. That student chooses another student and shows him/her how to use the program. The process continues until each student has experienced using the program and has shown another person how to use it. (To accommodate the last student, you may need to have that student show an aide or a student from another room how to use the system.)

2. Select a computer program as above, but this time begin with the computer system turned off. The goal this time is to help your students learn how to turn the system on and how to load and run a program. Each student is to master this process in a "learn by doing and learn by showing" environment.

   It is helpful to design a license or certificate for students who learn to use a computer at this level. The license might state that the student is fully certified to use the computer system and to show others how to use it.

3. Show your students how the computer library (both software and print materials) is organized and how to find a program that meets their needs. This activity assumes, of course, that you actually have a software library. The library might physically reside on tapes and disks stored in your school's media center. Students need to learn to use this type of library just as they learn to locate print materials in a media center. Instruction should be in the "learn by doing" mode. For example, a fifth grader should be able to go to the library and find a spelling drill program to fit his/her needs.

Exercises

1. We assume that by now you are reasonably skilled in programming in a language such as BASIC or Logo. Think back to when you were first learning to program. Describe the experience and its implications in helping children learn how to program.

2. Did you, as a high school or college student, study a foreign language? If so, compare and contrast that experience with the experience you had in learning to program. Your discussion should tie in with your language acquisition as a child and include educational implications.

3. The previous section mentions the idea of a "computer-rich" environment. Describe a computer-rich environment for elementary and middle school students. Are schools, homes and places of entertain-

ment all part of this environment? What role can parents play? What do you need to know to help your students have a computer-rich environment? That is, what level of teacher knowledge is essential?

Problem Solving

Education has as one of its major goals helping students learn to cope with problems of their current and future world. Not only are there an untold number and variety of current problems, but old problems change and new ones emerge as societies and technologies change. The student who is learning to cope with problems is aiming at moving targets.

For a given person, all problems can be divided into two categories. The first category consists of problems the person recognizes as ones s/he knows how to solve and can proceed to solve immediately. Generally, these are problems that have been previously encountered or studied. Computer scientists call such problems primitives. The second category is all other problems—those not immediately solvable by that person.

Our educational system has recognized the dual nature of problems. Rote memory, experience, drill and practice—all can help build a student's repertoire of primitives. The problems chosen to be mastered in this manner are usually the most frequent, the most important, or the building blocks useful for attempting to solve the second category of problems.

How can one respond to a problem which is not immediately solvable? Such problem solving is the essence of intelligence; it is being able to use one's senses to acquire data, one's brain to process the data, and one's body to take appropriate action. Problem solving skills improve through study and practice. Equally important, one learns to make use of and build upon the work of others. The accumulated knowledge of the human race is stored in its books, journals, films, recordings, tools and instruments. Students learn to make use of this accumulated knowledge.

There is a considerable difference between accumulated knowledge stored in a book and accumulated knowledge embodied in a tool or instrument designed to accomplish a specific task. By and large, print materials represent a static storage of accumulated knowledge, while tools and instruments represent a dynamic storage. A book about bicycles is different than a bicycle. A child can learn about bicycles by reading, but learns different things when actually acquiring riding skills on a "real" bike.

A watch is an instrument that falls between static and dynamic storage of accumulated knowledge. Inherent to the construction of a watch is a tremendous amount of scientific information about the measurement of time. A watch can be "read" to produce numbers representing time. But the reader must have knowledge to interpret these numbers, just as the reader of a book must have knowledge to interpret what is in the book. For a person who can read a watch, the watch is a most useful instrument.

A computer, like a watch, is an instrument, but a computer is also a general-purpose aid to problem solving. Like the watch, a computer spans the range from static to dynamic storage of accumulated knowledge. Anything that can be stored in a book can also be stored in a computer. But a computer can follow a detailed step-by-step set of instructions. A computer can act upon or act following the knowledge it contains.

A computer is like many of our tools or instruments in that little effort is required to use it to do some very useful things. Suppose that a computer has been programmed to draw a graph or chart of some data. You type in the data and out comes a graph or chart. What could be simpler? Or suppose a computer has been programmed to check the spelling of each word a secretary types. Such an aid to word processing is now in common use.

Books are associated with the skills of reading and writing, which require considerable effort to master. Similarly, computers are associated with skills such as computer programming, the design and analysis of algorithms, and hardware design and maintenance. It takes a substantial amount of effort to learn these skills.

Every academic discipline has problems whose solution can be aided or accomplished by a computer. There is a growing collection of high quality programs that solve or help solve the same types of problems we teach students to solve using pencil and paper or by other means. For example, middle school students learn to organize data and draw a pie chart to show percentages falling into each of several categories. The same data can be input to a computer which can output a nicely drawn, multi-colored pie chart.

The educational implications of growing libraries of computer programs are immense. If a computer can solve or help solve a particular type of problem, the question arises: What should students be able to do mentally and what should they be able to do with the aid of other resources such as pencil and paper? There are no simple answers to this question. But it is evident that educators must learn these computer capabilities so they can make reasoned judgments as they guide their students in acquiring a relevant, useful education.

Exercises

1. What is a problem? First, cite examples from a number of different areas. Next, give a general definition that encompasses these examples. Then try to think of counter examples: non-problems that fit your definition and problems that don't. Adjust your definition.
2. Look for and examine instructional materials used at the elementary and middle school level that define problem solving and which are designed to help students learn about problem solving. In what ways do students receive explicit instruction on problem solving? What assumptions does the instruction make? Perhaps you can suggest some changes that would lead to increased problem-solving abilities on the part of students.
3. Many calculators have a square root key. Various paper and pencil methods for calculating square root are taught at the secondary school level. In your everyday life, at home, work or school, do you ever need to calculate a square root using paper and pencil? Discuss the educational implications of this situation.

Steps in Programming to Solve a Problem

Computer scientists have carefully studied the steps to be followed in writing a program to solve a problem. The steps discussed in this section would be followed by a professional to produce a professional quality program. Elementary or middle school students learning to program should not be expected to follow all of these steps in detail at first. However, the ideas of these steps are important and they gradually should be made explicit to students as they develop programming skill. Note how much of each step is independent of any specific computer system or programming language.

STEP #1: UNDERSTAND THE PROBLEM. Whether a problem is to be solved with the assistance of a computer or by other means, the first step is always to try to understand the problem. Often it is the most difficult step. Most problems are not clearly defined. It doesn't make sense to start writing a program to solve a problem before one understands what the problem is.

Problem understanding involves obtaining clear specifications of what is known (what are the givens?), what is wanted (how can we tell if we have solved the problem?), and what are the ground rules, or allowable types of operations. These three aspects of a well-defined problem are worth remembering. Elementary school students can learn them.

Often problem understanding is enhanced by examining special cases or by working some examples by hand. For example, suppose that a dictionary has been stored on a computer system's magnetic disk and the problem is to write a program that accepts as input a word and produces as output the word's definition provided the word is in the dictionary. One can gain insight into this problem by looking up some words in a printed dictionary and observing what difficulties might be encountered. Added insight is gained by trying to look up a word that is not in the dictionary.

All of a person's formal and informal education and experiences contribute to the area of problem understanding. A person with no training in business is unlikely to be able to understand a business problem. A person who cannot read will have trouble with a problem presented in written form. A person with a limited vocabulary may not understand some of the words or ideas used in expressing a problem. A person who has not lived in our society may not understand our social values and social problems.

STEP #2: SELECT OR DEVELOP A SOLUTION PROCEDURE. Suppose one has a good understanding of the problem to be solved and believes that a computer will be a useful and appropriate aid. The next step is to select or develop a solution procedure.

The term "select" suggests that one has a repertoire of procedures available and merely needs to select an appropriate one. Consider the following example:

Problem: Pat goes into the candy store. A small bag of candy costs 18 cents. Pat wants to buy four bags of candy. How much will four bags cost?

Some first grade students and most second grade students can read and understand this problem. What procedures are available to such students? We rule out

procedures such as asking the clerk. The student is probably fairly good at counting, and can do simple addition and subtraction problems. After some thought the student might decide that the problem could be solved by adding 18 to 18 to 18 to 18. But this is probably beyond the student's formal addition skills. Thus, the student is led to select counting as the procedure to be used. The first bag is 18 cents. Starting at 19 cents the student counts the second bag, hopefuly arriving at 36 cents. The counting process continues until the money needed for four bags has been counted and the problem is solved.

Now consider a fifth grade student faced by the same problem. The student can count, add, subtract, multiply, divide, draw a bar graph, and do many other things related to solving math problems. The selection task is more difficult because of the wider range of choices. The thought may pass through the student's mind that the problem can be solved by counting, by $18+18+18+18$, by 4×18, and by $4\times20-4\times2$. The student's wider knowledge means there are more right ways and more wrong ways to attempt to solve the problem.

The selection process is essentially one of matching the given problem to a primitive, basic model or to a pattern with which one is familiar. The second grade student is familiar with money and the counting of money. Thus this student is led to select counting as a solution procedure. The fifth grade student is familiar with more complicated primitives such as multiplication and division. This student is also familiar with models such as "Total cost equals number of units purchased times cost per unit."

Increasing students' storehouse of basic models and primitives may increase their overall level of problem solving ability. More of the problems they encounter will match their knowledge and can be solved by a simple selection and then execution of a procedure. However, the selection process becomes more difficult as more models and primitives are added. Also, presumably one studies the most useful models and primitives first, eventually arriving at a state of studying those that are useful only very infrequently. Therefore it should be evident that acquiring more and more primitives does not necessarily improve one's problem-solving skills, and may not be the best use of one's educational time. Rote memory is an essential part of problem solving, but is only part of the total answer.

No matter how much one memorizes, many of the problems one encounters are not primitives—they are not immediately solvable. Certain general methods have been developed for attacking such problems. Two ideas used by all computer scientists are stepwise refinement and subroutines.

In stepwise refinement, one systematically breaks a problem into smaller, more manageable pieces. Figure 1 illustrates this idea. If one can solve Subproblems 1, 2, and 3, then the original problem can be solved. If a particular subproblem is not immediately solvable, further stepwise refinement is necessary. The goal in stepwise refinement is to break the original problem into primitives.

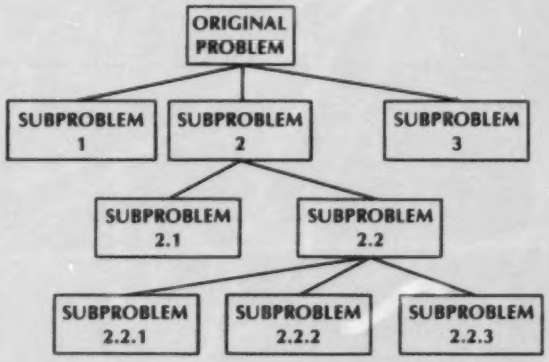


Figure 1. Stepwise refinement.

A subroutine is a self-contained piece of a program designed to accomplish a specified task. Often the task is one that must be accomplished many times in the process of solving a larger problem. For example, suppose you want to write a program to draw a flower garden. You might develop a subroutine to draw a daisy, a second subroutine to draw a rose, and a third subroutine to draw a cloud. The picture will contain many daisies, roses and clouds.

It is possible to develop subroutines useful in a variety of programs. Writing block letters of a specified size on the computer display screen is an example. A library of such subroutines is an invaluable aid to the serious programmer.

STEP#3: WRITE THE PROGRAM. If a problem is simple enough, then steps one and two can be done mentally. Writing a simple program is no more difficult than composing a few sentences to form a short paragraph.

The more complex the problem, the greater the amount of preliminary work before programming actually begins. For a really complex problem, the preliminary analysis and representation of a procedure to solve the problem may be three-fourths or more of the total problem-solving task. For such problems, the development and representation of a solution procedure is done in detail before one begins programming.

At one time it was felt that a flowchart was a very good way to represent a procedure to be programmed. In the late 1960s and early 1970s, a number of books were written on flowcharting. But in recent years it has become clear that flowcharting has considerable limitations and the teaching of it has diminished.

In place of flowcharting, or to supplement relatively crude block diagrams, computer scientists now use a natural language, stepwise refinement. This can be written like the outline of an English composition theme.

In any event, much of the thinking for solving a problem is completed before one begins to write program statements. Indeed, on large programming projects a programmer-analyst may carry out steps one and two, while lesser paid people such as programmer-trainees carry out the third step.

Generally the choice of what programming language will be used is made during the first or second step. Sometimes the choice will be dictated by the workplace situation. For example, a company may happen to do all of its programming in COBOL, or the computer system to be used may only support the language BASIC.

Each language has distinctive characteristics that make it particularly well suited to certain tasks. The professional programmer is apt to be fluent in several programming languages, and the choice of a language may be dependent upon the particular procedure to be used to solve the problem.

STEP #4: PROGRAM AND DATA ENTRY. Many computer programs are designed to process a set of data. Data to be processed by computer must be in computer readable form. So part of solving many problems with a computer is data preparation and data entry. Also, the computer program itself must be put into computer readable form. Typically this is done by typing it into a computer memory using a keyboard terminal or by keypunching the program onto cards. In a large-scale computer programming operation, data entry and program entry tasks will be done by professional data entry clerks. They tend to be better typists than programmers but receive less pay.

To a considerable extent, however, computer programmers do their own program and data entry. Having good typing skills saves a lot of time and reduces errors. Few elementary and middle school students are good typists, and most take a very long time to type a small amount of information into a computer.

STEP #5: DEBUG PROGRAM, AND VERIFY ITS CORRECTNESS. After completing program and data entry, the programmer is ready to run the program on a computer. The program usually will contain errors. An error in a program is called a bug. Debugging consists of detecting and correcting errors in a program.

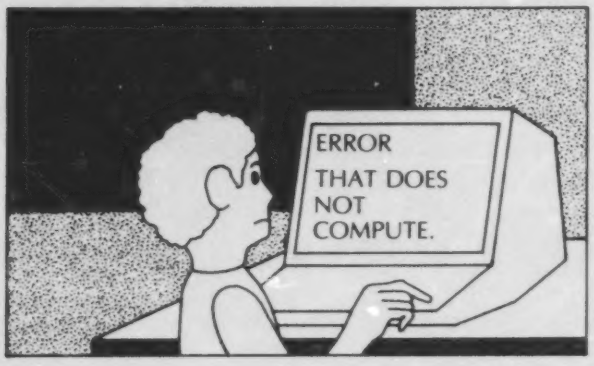
All programmers make errors. A good programmer generally makes fewer errors than a poor programmer, and a good programmer is skilled at detecting and correcting errors. Fortunately, skill in debugging programs increases with practice, and the computer itself provides some assistance. Programming languages have very precise rules, called syntax rules, that specify exactly how programming statements must be written and punctuated. Any violation of these rules is called a syntax error. The computer program that translates from a high level language such as Logo, BASIC or COBOL into machine language also checks for syntax errors. It outputs error messages indicating the nature and location of errors it detects.

More difficult to detect and correct are errors in program logic. These errors cause the program to perform in unexpected ways, failing to solve the problem the program was designed to solve. One aid to detecting errors in program logic is program testing. This involves comparing what the computer produces with some results that are known to be correct. These results may have been produced by a different program or by hand. If the correct results and computer results agree, one can have increased confidence in the correctness of the program.

It is difficult and sometimes nearly impossible to verify that all parts of a long program are correct and that the program will perform correctly in all possible cases. Indeed, this is currently an important research topic in computer science.

Some concepts of debugging and testing procedures can be taught at the elementary and middle school levels and do not require access to computers. The Applications at the end of this section give examples.

STEP #6: COMPLETE PROGRAM DOCUMENTATION. Program documentation consists of a careful statement of the problem, flowcharts or natural language procedures written in an outline form, sample computer runs on sets of test data, a listing of the program, and directions on how to use the program. The process of documenting a program begins with the process of clearly defining the problem to be solved. The solution procedure is part of the documentation, as is a discussion of the testing procedure used to help verify program correctness.



Program documentation serves a number of purposes. It generally consists of several parts, designed to communicate to different groups of people. One part of the documentation may communicate to the management level people who originally stated the problem to be solved. They want written assurance that the program solves the problem they had in mind.

Another part of the documentation may be directed to users of the program. Often they are not programmers and have little insight into how a computer functions. They must be given precise directions on how to prepare data for processing by the program and on how to use the program. They may need information on how to read and interpret the computer output.

Finally, there is documentation directed towards computer programmers—the programmer who wrote the program and programmers who may need to modify the program and/or correct errors that may be detected sometime in the future. This documentation is usually quite technical, as it represents a programmer attempting to communicate to another programmer precisely what the program does and how it does it.

In a large professional programming operation there may be technical writers whose sole task is to help document programs. However, most programmers have to do their own documentation. Thus, it is important that programmers have good technical writing skills.

Application 1 (Giving and Following Directions)

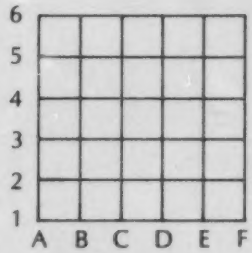
A computer program is a set of directions that a computer can interpret and execute. A major aspect of computer programming is writing precise sets of directions. The exercises below will help students improve their skills at giving and following sets of directions.

1. Have each student take a blank sheet of paper and place his/her pencil near the center of the paper in a ready-to-draw position. Select a simple line drawing and keep it hidden from the student's view. Then "tell" the students how to copy the line drawing. Examples of instructions might include: "Draw a horizontal line 3 cm. in length. Now, starting at the right-hand end, draw a line slanted 45 degrees up and to the right, 5 cm. in length. Etc." All of this is to be done without the use of a ruler.

   Initially this game should be played with very simple drawings. Students can be asked to identify what is being drawn. Later, students can select or make up drawings and take the role of teacher. Or students can work in small groups so that more students can take the leadership role.

2. Repeat 1 above, this time using graph paper marked off in 1 cm. squares. You will notice an amazing improvement in the quality and recognizability of the results. The graph paper improves the direction-giving language. This exercise gives some insight as to why some programming languages are better than others.

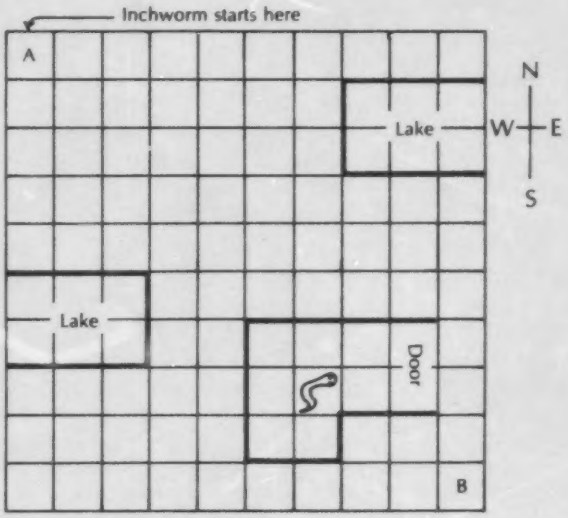
3. A more precise and easy to follow set of directions can be given using a coordinate system on graph paper.



   Each intersection point on the grid can be uniquely identified by a pair of coordinates. Thus A1 is the lower left hand corner. An instruction such as "start at A1 and draw a straight line to C5" leaves no room for ambiguity. It communicates precisely, and every student following such a direction should produce excactly the same line.

   Students can be given graph paper and asked to label the coordinate axes. Then they can be asked to follow one or more sets of directions given by the teacher. Finally, they can be asked to write down a set of directions to draw something. Papers can be exchanged so that all students get a chance to have their sets of directions executed.

4. A variation on the above types of activities is a game called Inchworm, which is suitable for quite young students. A small creature, called an inchworm, is to be moved from one spot to another on a squared grid.



   A program for the inchworm consists of a sequence of directional specifications, such as EESSSSEESW.

The specification E means the inchworm is to move one square east. The goal is to write a program that moves the inchworm from some starting point, say square A, to some ending point, say square B. Of course one should avoid drowning the inchworm, or having it fall off the board. An inchworm program needs to be debugged and tested, much like a computer program.

After students master the initial game, one can add two more instructions. U specifies the inchworm is to pick up an object, and D specifies the inchworm is to set the object down. Now an object can be placed someplace on the grid, say in the house. The inchworm is to be directed to the object's square, to pick up the object (U), to move to some destination, and to set the object down (D).

The concept of looping or iteration, which is very important in computer programming, can be illustrated using the inchworm. A pile of ten objects is located on the grid. The problem is to move all of these objects to another specified location, when the inchworm can only carry one of them at a time. After students have written a certain set of steps ten times, you might suggest an abbreviation, such as "Repeat the following sequence ten times."

Application II (Robot)

A robot is a computer-controlled machine that can carry out a variety of tasks automatically. Most people's image of a robot is a walking, talking, metal, human-like creature with an intelligence similar to that of a human. While a few human-like mechanical robots have been built, none have the intelligence or most other characteristics of a human.

However, robots are now quite standard in modern automobile assembly plants and other manufacturing and assembly operations. These are computer-controlled devices programmed to perform a particular assembly operation such as weld two parts together, screw a nut onto a bolt, or operate a spray painter. Changing the robot's program "teaches" the robot to perform a different task. This is particularly useful in industries like automobile manufacturing where there are periodic model changes. The activities below are designed to give students insight into the capabilities and limitations of robots.

1. Begin by selecting a student to give directions and by having the teacher serve as the robot. The robot should initially be seated at a desk with a pencil in its hand. The student is to direct the robot to move to the pencil sharpener, sharpen its pencil, and return to a seated position at the desk. See 2 below for some of the types of instructions that might be given.

   The activity is most instructive and amusing if the robot interprets each instruction with little regard for the consequences. The robot will run into furniture or into the wall, be unable to insert the pencil into the sharpener, and so on. If it is impossible to understand or to carry out an instruction, the robot should respond "That does not compute!" One purpose of this is to begin to understand the need for precision in specifying and executing instructions.

2. The class working together should decide upon a set of allowable operations. Each is a "primitive" that the robot is to be able to understand and execute. A typical set of primitives might be:

   a. Pick up pencil from desk.
   b. Set pencil down on desk.
   c. Stand up.
   d. Sit down.
   e. Turn right a specified number of degrees.
   f. Take a specified fraction of a step (up to one full step) forward.
   g. Take a specified fraction of a step (up to one full step) backwards.
   h. Insert pencil in sharpener. (This can only be done if pencil sharpener is within easy reach.)
   i. Turn handle of sharpener clockwise three times.
   j. Remove pencil from sharpener.

   Now 1 above can be repeated, with students taking turns being the robot and the robot programmer. Any instruction other than a-j above should cause the robot to respond, "That does not compute!"

3. Suppose the robot can only understand the types of operations a-j given in 2 above. Can the robot be directed to go to a particular student's desk, pick up that student's pen, and place it on the teacher's desk? Can the robot be directed to empty the pencil sharpener into the wastebasket? Questions such as this give insight into the difficulties in designing a programming language—in this case the types of instructions the robot can understand.

   Make up a list of several tasks such as sharpening a pencil, emptying the pencil sharpener into the wastebasket and feeding the goldfish. Have each student make up a robot-programming language that s/he thinks will be suitable for telling a robot how to accomplish all these tasks. Then let students try out their languages on each other for specific tasks.

   Some students will add more special purpose instructions to their language such as "pick up fish food." Others may try to add general purpose instructions directing hand movements. A hand can be opened and positioned correctly and then instructed to grasp the object between its open fingers.

Application III (Programming and Debugging)

In the robot activities given above, the person directing the robot did not have to figure out the entire set of directions in advance. Rather, the programmer could continually examine the robot's location and give instructions to overcome difficulties, such as a chair in

the robot's pathway, as they occurred.

Consider a robot that can execute the following types of instructions.

a. Stand up (from a seated position).
b. Sit down (assumes a chair is immediately behind the robot).
c. Take a specified fraction (up to 1) of a step forward.
d. Turn to the right a specified number of degrees (not to exceed 360 degrees).
e. Extend right arm forward and parallel to the floor until it touches something or until extended to its full length.
f. Lower right arm to robot's side.

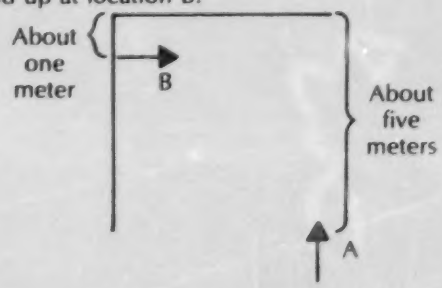
Suppose that the robot has a sense of touch in its right hand and can make a decision based upon the outcome of executing instruction e. Thus, execution of e can be followed by an instruction such as, "If something was touched, go to instruction 8. Otherwise, continue with instruction 4." Activity 1 below illustrates what this added capability allows.

1. Suppose the robot is seated facing a wall that is about 5 meters away, and that there are no obstacles in the robot's path to the wall. What will the following program accomplish?

Step #	Instruction
1.	a
2.	c, 1/2 step
3.	e
4.	f
5.	If no touch (in step 3) go to step 2. Otherwise, go to step 6.
6.	d, 180 degrees.

The robot will end up close to, but facing directly away from, the wall. Notice the abbreviations used in writing the program steps. This is much easier than repeatedly writing out long instructions. Give the class several examples of programs involving decision making, and allow each student to gain experience in reading and following such a program. Notice that the robot "remembers" the results of its most recent hand extension. It need not use that information right away.

2. Consider the picture given below, where the arrow represents the location of a robot and the direction it is facing. The robot starts at location A, several meters from each wall. The goal is to have the robot end up at location B.



a. Have each student write a program that is supposed to move the robot from A to B.
b. Select a student's solution that you are sure is incorrect. Or, use a "solution" from "some other class" that you know is incorrect.
c. Have one of the students execute the incorrect solution.
d. Have the class, working together, debug the incorrect solution.
e. Have the students reexamine their own programs and mentally carry out the steps. In this process it is okay to use pencil and paper. This is called "desk checking" and is a very important aspect of program testing and debugging.
f. Have students exchange papers, and desk check each other's programs. The papers are then to be returned, and any errors detected are to be corrected.

3. Add the following capability to the robot:

g. Use your eyes to search for any obstacles located within one half step forward.

After g has been executed, the robot can carry out instructions such as "If obstacle observed, go to step 5. Otherwise, go to step 12." Repeat Activity 2 with this new robot, but allowing for a possible obstacle someplace in the way.

Application IV (Debugging)

Recall that a procedure is a detailed step by step set of directions that can be mechanically interpreted and carried out by some agent. It is designed to solve a problem or type of problem. But there is no guarantee of success, and a procedure may contain errors (bugs). Detecting and correcting errors is an important aspect of problem solving.

1. A student has "solved" the following subtraction problems. See if your students can detect the type of error that is being made, describe it, and produce correct answers. If these calculations are not suited to the level of your students, make up some that are.

17	45	193
−8	−28	−65
19	27	138

2. Play a tune on the piano which is familiar to students, but insert one or more errors. See how the students react to these errors. Have them help debug your attempt to play the tune correctly.

3. Bake your students some bread with the salt or some other ingredient missing. A recipe is a procedure. See if your students can detect an error in your recipe. Discuss with the class how one might debug a recipe.

4. Errors in spelling and grammar are bugs in writing. Give students a paragraph that correctly conveys a particular idea but which contains errors in spelling and grammar. Have the students detect and correct the errors.

Exercises

1. Debugging is an essential feature in the overall process of writing a computer program to solve a problem. In what sense does a musician composing a new piece of music use debugging ideas? Discuss the composing/programming analogy. Then make up and discuss an analogy between programming and writing.
2. Mathematics, especially arithmetic at the elementary and middle school levels, is often taught on an "all right or all wrong" basis. The concept of debugging a partially correct procedure may not be included in the curriculum. Suggest several examples of how more debugging ideas could be introduced into mathematics instruction. How do you think the increased emphasis upon debugging ideas will affect student attitudes and success in math?

CHAPTER SEVEN

COMPUTER AND INFORMATION SCIENCE

A computer scientist studies both the underlying theory and the practical applications of computers. Thus, the people who designed and built the first general purpose electronic digital computers in the 1940s were certainly computer scientists. But even before that time there were a few people who devoted their professional careers to studying the capabilities and limitations of automatic computing machinery. Various mathematicians and engineers made significant contributions to the foundations of computer science in the 1930s.

Once mass production of computers began, the number of computer scientists grew rapidly. By the late 1950s, most major universities had computers. They were often used by scientists, engineers and mathematicians for research and instruction. Enrollment in computer-oriented courses grew rapidly.

Gradually it became evident that computers are useful in all academic areas and that educators in all disciplines could learn to make effective use of computers. The idea emerged that a computer is an information processing machine. One outgrowth of this is that many colleges and universities now use the title "Computer and Information Science" to denote the department that conducts courses and research in the computer field.

The development of computer and information science was greatly helped by the formation of professional societies devoted to various aspects of the field. Among these were the Association for Computing Machinery (ACM), the Data Processing Management Association (DPMA), and the Institute for Electrical and Electronics Engineering (IEEE) Computer Society.

The early contributions of the Association for Computing Machinery (founded in 1947) are especially noteworthy. The ACM published detailed recommendations for an undergraduate computer science curriculum with the title "Curriculum '68" in 1968. Over the next ten years the "Curriculum '68" recommendations helped guide the writers of textbooks and the developers of courses as the teaching of computer and information science became common in colleges and universities.

In recent years, "Curriculum '68" has been updated to "Curriculum '78" and both the DPMA and IEEE have published curriculum recommendations. To a large extent such curriculum recommendations serve to define the field of computer and information science. They also serve as guidelines to the 400 or more colleges and universities in the United States that offer the bachelor's degree in some aspect of the computer field.

In this chapter we discuss four major subfields of computer and information science. These are modeling and simulation, information retrieval, computer graphics and artificial intelligence. Each is of growing importance and has direct bearing upon education at all levels.

Modeling and Simulation

As a child you may have played with and built toy models of cars, boats, trains, airplanes and houses. A toy model has some of the features of the "real thing." Some models are built exactly to scale or have other characteristics that are precise representations of the real thing. Sometimes such scale models are used by scientists. An exact scale model of an airplane wing can be studied in a wind tunnel to gain insight into how such a wing will perform on a full scale airplane. Architects build scale models to help their clients see proposed building designs.

If one had to give a very short definition of science, a good answer would be, "Science is description and prediction." Scientists strive to accurately describe how things are. They strive to understand relationships so they can predict what will happen under a given set of circumstances.

Science makes use of scale models for both description and prediction. But scale models are not adequate for many applications. Suppose, for example, you wanted a scientific description of our solar system. A scale model would help. It could show how an eclipse of the moon or sun could occur. But it could not be used to make accurate predictions of when a

future eclipse would occur. Much of what research scientists do is to create new models and to try to prove their models are "correct." That is, they try to prove their models are more accurate at description and prediction than are previous models, and that their models conform to our current knowledge of the physical world.

Astronomy is particularly interesting because of the variety of models that have been suggested over the years. A model that was held to be correct for many years pictured the earth as the center of the solar system, with the sun rotating around the earth. It took considerable careful scientific observation to show that this was not an accurate model. We must remember that there is a difference between a model and the object or event being modeled. There can be many different models of the same physical object or system.

Mathematical models provide a useful alternative or supplement to scale models. Mathematical models are often very easy to develop, easy to use, and easy to change. In fact, you use mathematical models all of the time without even realizing it. You are thinking about going out to buy candy bars for yourself and a friend. You know that candy bars are 35 cents each. You look into your billfold and see three quarters. You conclude you have enough money, and set out to buy the candy.

The notation "35 cents" is a mathematical model, as is the word "quarter." In all of your thinking process, no money or goods changed hands. The mathematical models allowed you to mentally simulate the entire transaction.

How old are you? When were you born? What is your weight and height? What is the current date? In all cases the answers are numbers, mathematical models of certain aspects of the world.

It is likely that you know several mathematical formulas.

Distance = Rate × Time	D = RT
Interest = Principal × Rate × Time	I = PRT
Area = Length × Width	A = LW

A formula is a model, giving a precise statement about how certain variables are related. The models created by scientists almost always involve formulas. That is one reason formulas are an important part of mathematics education.

A simple model such as D = RT is adequate to describe a simple situation. But progress in science and engineering has led to the study of more and more complex problems, and to the need for more complex models. Thus, mathematics has grown in importance, along with the use of computers to carry out complex mathematical calculations.

Weather forecasting provides a good example. Scientists know a great deal about the theory of weather and how to predict it. But the amount of data needed to describe the current weather is immense, and the computations that must be applied to that data in order to predict future weather are also immense. Therefore, the weather bureau uses very large computers. The use of these computers and increased knowledge of the theory of weather have led to substantial improvements in weather forecasting.

Every field worthy of the name science uses models and continually strives to make these models more accurate. Once an acceptable model has been developed, it can be used to study alternative futures. That is, it can be used to simulate what would happen under a given set of circumstances. If a computer is used to carry out the computations in the simulation, the overall process is called a computer simulation.

Let's look at an example. Suppose that you are a traffic engineer in a city that has bad traffic problems. You have some ideas on how to solve these problems. You think that if certain key streets were made one way and if the timing of certain traffic lights were changed, then the problems would be solved. However, it would cost a lot of money and would take a lot of time to try out your "guess" at a solution. And maybe your guess would be wrong.

Suppose instead that you had a good mathematical model of traffic flow in your city. It would take into consideration each major street, the traffic lights, and data about the source and destination of the traffic. Suppose the model were computerized so you had a computer simulation.

With the computer simulation you could change the street directions and traffic lights (in the computer) and see what happens. You could increase the number of cars entering the system to see what happens as population increases. You could do repeated trials; trial and error is an important tool in problem solving. The cost and time to perform each computerized trial would be very small relative to actually making the physical world changes.

As you can see, computer simulation is a very valuable tool for problem solving. It requires a good model plus adequate computer facilities. Developing a good model requires very good knowledge of the underlying theory (the science) of the situation to be modeled. But the model by itself is not adequate. The model must be programmed for a computer, and the computer must be adequate to the computational task. The model and the computer together make computer simulation a major tool of modern science.

Applications

The elementary and middle school curriculum is rich in activities that involve modeling and simulation. The activities in this section can be used to help students come to understand the ideas of modeling and simulation.

1. You may not realize it, but much of the mathematical content of elementary education is modeling and simulation. Consider the following problem. "Jane goes to the store to buy some apples. Apples are $ .25 per pound. She buys six pounds. How much money does she spend for the apples?"

   There are two obvious mathematical models that a student might use to solve this problem. The first is addition.

   Total cost = Cost for 1st pound + cost for 2nd pound + cost for 3rd pound + etc. + cost for 6th pound

   The second model is multiplication.

   Total cost = Cost per unit × number of units

   The point is, in order to solve this problem, the student must select (or develop) an appropriate mathematical model.

   Then the student carries out a computation and arrives at an answer. Is any money actually spent? Are any apples actually purchased? Obviously not. The whole problem is a simulation. The student solving the problem mentally imagines the process of Jane making a purchase. The model and the computation involved in this problem are simple enough so a calculator or computer is not needed. But both the modeling and the simulation process are illustrated.

   Select some word problems your students can solve. Determine some possible mathematical models that fit these word problems and are appropriate to your students. Discuss with your students the nature of mathematical modeling and the role it plays in solving word problems. Then give the students some problems in which they are to specify the mathematical model. Encourage them to give at least two different models for each problem.

2. Many games can be considered to be models of certain aspects of the real world. Playing such games is simulating those features of the real world. The game of Monopoly is a good example. It was modeled after certain aspects of business in Atlantic City, New Jersey.

   Many children enjoy such games and will play them hour after hour. In playing Monopoly they will be buying, selling, paying rent, collecting money—all of which involve working with numbers. They will be plotting strategies, trying to make advantageous trades with their opponents, and so on. They gain experience in solving problems that occur in the game; they also have the social interaction of playing the game.

   There can be considerable worthwhile learning going on even if children are playing games just for fun. But learning can be enhanced if the teacher points out various aspects of problem solving and the way the games are modeled after certain aspects of the real world.

   Select a game such as Monopoly and discuss with the class how it can be considered as a model of part of the real world, and how playing it is simulating certain aspects of the real world. Have students select and play other games of interest to them and then have them explain the games in terms of modeling and simulation.

3. The branch of mathematics called probability can be used to predict future events. If we flip one coin it will land with either a "heads" or a "tails" showing. Each is about equally likely so mathematical theory doesn't help in predicting the outcome. But if we flip 10,000 coins we cay say with considerable confidence that approximately 50% will show heads and 50% tails. That is, with very high probability the number of heads will be between 4,500 and 5,500.

   Elementary and middle school students usually do not have good insight into the underlying probability theory of tossing a pair of dice. If one thinks of the two dice as quite distinct, then any one of 36 outcomes is possible. They can be listed as ordered pairs (first die, second die). Thus, the possible outcomes are (1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), etc. to (6,6). If the dice are "true" (not loaded), then each of these 36 outcomes is equally likely.

   By studying these outcomes you can see that there are six ways a double can occur, and six ways that a total of seven can occur. The probability of throwing doubles is exactly the same as the probability of throwing a seven. This means that if a pair of dice are thrown a large number of times, approximately equal numbers of doubles and sevens will result.

   Dice throwing experiments can be carried out in the classroom. Students can keep detailed records of the outcomes. From these records they can develop predictive models for what to expect if a large number of dice tosses are tried in the future. Note that there is no guarantee that a particular pair of dice are true. Thus, a student's model may predict that the ratio of sevens to doubles will be about 1:2. This might be a good model for a particular pair of dice. What is important in this type of activity is that the student be led to create an empirical model.

4. Encourage students to see that models are used in everyday life. Have them make lists of models. They can compare and contrast the models with the things being modeled. Some examples include:

   - A globe (map) of the world, or a highway map.
   - A person's name.
   - An artist's painting of a landscape.
   - A poet's description of a person in love.

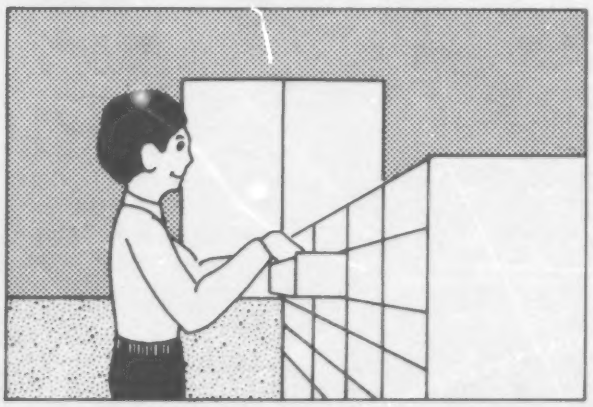
- A printed sheet of music.
- Paper money and coins.
- A hand-drawn circle or square.
- A printed TV schedule.

By studying, discussing and thinking about such models, a student can understand that the model and the "real thing" are not the same, but that models are indispensable in our lives.

Information Retrieval

Learning to "look up" information is one of the most important aspects of education. One learns to use a dictionary to retrieve the precise meaning and pronunciation of words; an atlas to retrieve information about the location of cities and countries; and an encyclopedia to retrieve information about people, events and things.

Information can be stored in a computer and accessed by use of computer terminals. The field of computerized storage and retrieval of information is called information retrieval, or IR for short. IR is a very important and rapidly growing part of computer and information science. Undoubtedly you are aware of computerized airline and motel reservation systems, computerized police and FBI files, computerized social security and Internal Revenue records. Perhaps your public library makes use of computers; the Library of Congress makes extensive use of them.



There are two general categories of computerized information storage and retrieval systems: data base systems and bibliographic systems. Each of these will be discussed.

The data base system gets its name from the fact that it contains a fixed base of data—one can know exactly what types of information are stored. A good example is provided by the student records for a college or university. For each student the school keeps the same type of information, including name, address, list of current courses, and a list of courses and grades from previous terms. The records also contain parents' names and addresses, student's major, student's year in school, and so on.

Only a limited number of types of data is being kept. For example, it is unlikely that the student records system can tell you if Mary Jones has ever worn braces or if George Smith has ever had a broken leg. But the system can be used to determine a student's intended major, total number of credit hours obtained to date, and grade point average.

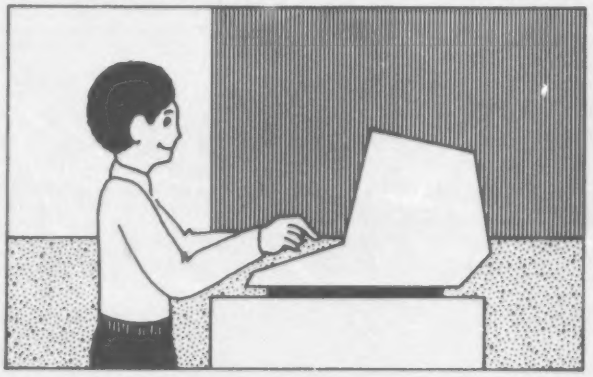
There are many computerized data base information retrieval systems in use. Examples include airline reservation systems, hotel and motel reservation systems, property tax systems, state and Federal tax systems, stock market quotation systems, police and FBI files, and certain types of medical files.

A computerized data base system consists of a base of data, computer programs that make it easy to get the answer to certain questions, and computer programs that make it easy to add data and/or change the data. The data in the system may be changing quite rapidly. For example, the data base for an airline reservation system changes every time a ticket is sold.

Data base IR systems are often accessed via a network of terminals at widely scattered locations. Some systems can be accessed from throughout the country or world. The information can be updated and/or corrected rapidly. Information can be retrieved very rapidly, with speed of access being relatively independent of how close physically one is located to the data base. Stock brokers throughout the country have equally good access to transaction records for the New York Stock Exchange.

The data base can also be used to do research—to try to answer general questions related to the data. An airline reservation system can be used to look for patterns in increased or decreased usage at certain times of the year. A police information system might be used to look for patterns in locations or methods of crime. A student record system could be used to look for relationships between student grades and type of housing accommodation. Perhaps students who live in dorms get better grades than students who live in off campus housing.

Thus, a data base system is designed to answer some questions very rapidly and to allow research into other questions. The research questions usually require that a special program be written to examine and analyze the



data. Often the researcher is searching for patterns or relationships in the data. When large masses of data are involved, computers are an essential aid.

A bibliographic information retrieval system is considerably different. One can think of it as the computerization of some general aspects of a library. An example will help illustrate this.

Suppose you are a doctor, and you think that your patient has a certain rare disease. You would like to know the latest research results and treatment for this disease. You have a bibliographic information retrieval problem.

The problem is not easy. There are several hundred thousand medical articles published each year. Many are in foreign languages. You subscribe to a few of the journals, but don't have time to read them thoroughly. Your hospital subscribes to several hundred journals. A medical research center located a few hundred miles away subscribes to several thousand medical journals. So how can you find the needed information quickly and cheaply?

What is needed is a computerized bibliographic information retrieval system. You would like to be able to use a computer terminal to type in the name of the disease, or perhaps just its major symptoms. You would like to receive back a list of current articles and brief summaries of the articles. After reading the titles and summaries you may want to retrieve several complete articles.

All of this is now possible, although it isn't completely computerized. The field of medicine has been divided into a number of subfields. Each subfield has been assigned to one university or medical research center in the United States or in one of the other countries cooperating in the project. Each center subscribes to all journals related to its part of the field of medicine. The centers employ people who read, index and abstract each article. All of the indexes and abstracts are combined to form one large computerized bibliographic information retrieval system. It contains millions of abstracts and is growing very rapidly.

The actual articles are not available in computerized form. Thus, if you find an article you want to read, you must look it up in a library, or request that a copy be mailed to you.

The bibliographic system differs from a data base system in that users do not know in advance whether their questions will be answered. There may be no articles on the desired topic, or there may be hundreds. The articles might not contain the desired information.

Bibliographic IR systems are now commonplace. A college or university research library is likely to be able to provide access to hundreds of different systems, each with its own base of published materials. Of particular interest to educators is the Educational Resources Information Center (ERIC) system. ERIC is specifically designed to help educational researchers gain access to recent educational research results.

There are a growing number of IR systems that are both data base and bibliographic in nature. A single IR system might contain both abstracts to all articles in several leading newspapers or magazines, and detailed sports statistics for current and past games.

It is now possible to store entire articles and books in a computer system. This full text storage is becoming more economically feasible as the costs of secondary storage continue to drop. An example is found in the field of law. It is very difficult to index laws, law articles and law cases so that one can easily look up exactly the references desired. The smallest detail in a past case might prove to be important. So full text information storage and retrieval is used.

This means that the full text is stored, and computer searches are done on the full text of laws, articles and cases. Efficient full text storage and searching is an interesting and challenging aspect of computer science. It is clear that this is a growing field. It is also clear that it will eventually have a major impact upon the publishers of research journals. Even now a few journals are "published" by just entering the articles into a computerized system.

Applications

1. Some of the ideas of a data base and/or police information retrieval system can be illustrated by the following activity. Have each student in the class fill out a card not containing the student's name, but containing the following information.

   a. Sex.
   b. Height, rounded to nearest 2 cm.
   c. Weight, rounded to nearest kg.
   d. Age, rounded to nearest year.
   e. Hair color.

   Unless there are identical twins (or triplets) in the classroom, it is highly unlikely that two students have exactly the same data.

   Now take the entire set of cards and select one at random. Have all members of the class stand up. Then go through the list of the one student's data, one item at a time, directing students who don't satisfy it to sit down. For example, suppose the card you selected was a female of height 96 cm. You would first direct all males to sit down. Then direct all people of height 94 cm or less, or height 98 cm or more, to sit down. Continue the process until only one person remains standing. You have identified this person from the descriptive data!

   Data sets like this can be used for a number of activities. Consider the problem of finding the two students in the class who are most nearly the same. Suppose no two are exactly the same. How can you define "most nearly the same" in a manner that will allow the data base to be used to solve the problem? The police may have a partial description of a suspect, or several descriptions that don't exactly agree. How can they examine their files to find the names

of people who approximately fit the description? Set the class to work on these questions.

2. A major key to bibliographic information retrieval is indexing. Many school books contain an index so one can use these to teach students how to use an index. But how does one create an index? Select an article that is at a suitable reading level for the class. Number each paragraph. Then have each student read and index the article, using paragraph numbers rather than page numbers. Thus, if the article is about United States presidents and George Washington is mentioned in paragraphs 5, 9, and 23, the index would include:

   Washington, George, 5, 9, 23.

   You should observe that most students will agree that certain words belong in the index, but that there will be considerable disagreement on others. Also, people make errors in indexing. This exercise can give good insight into the difficulties of indexing, and the limitations of an index. Good indexing is a key to computerized bibliographic information retrieval.

   The activity can be followed up by a visit from your school's media specialist. S/he can answer many important questions such as: How is a library organized? What is the Dewey Decimal System and the Library of Congress system? Why are there two widely used classification schemes?

3. A telephone book contains information like one might find in a computerized data base information retrieval system. With the aid of your students and/or fellow teachers you can collect enough old phone books so that each class member will have one. Students can learn:

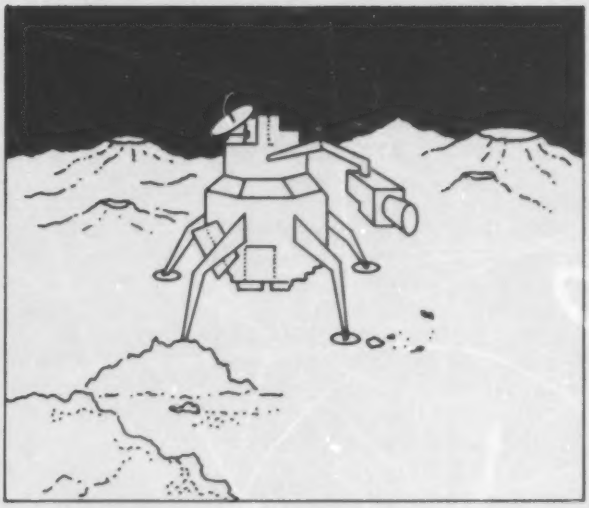
   - How to look up a phone number.
   - How to look up an address.
   - What an unlisted number is.
   - About the problem of people having identical names.
   - What is in the Yellow Pages.

   Suppose you know a person's phone number but not their name. Is a phone book a useful aid to finding the name? Would it help if the phone book were computerized?

Computer Graphics

It is often said that a picture is worth ten thousand words. Computer scientists are intensely interested in the use of computers to process picture-type information. It is likely that you are familiar with some of the progress that has been made.

Space probe pictures being sent back from Venus, Mars and Saturn are computerized and computer-enhanced. Similarly, the weather satellite pictures shown on television forecasts are processed by computer.



More down to earth, however, is the output display on a typical microcomputer system. The display unit is a television set and the computer system has some graphics display capabilities. The use of computer graphics is now commonplace.

Insight into computer graphics is provided by thinking about pen and ink plotting mechanisms. The idea of a machine-driven plotting device is much older than computers. You have probably seen a recording barometer or a recording thermometer. A clock-driven drum has a pen mechanism that is attached to a barometer or thermometer. The pen records the air pressure or temperature, automatically producing a graph over a period of time.

A plotter is a very commonly used computer output device. Under computer control a pen can be moved horizontally and vertically over a piece of paper. The pen can be raised and lowered. The whole setup is much like a "connect-the-dots" mechanism. A computer program tells the plotter mechanism the exact location of the dots to be connected, and when to raise and lower the pen. The dots to be connected can be quite close together, and the mechanism connects the dots at a very rapid rate.

Many modern computer-driven plotters contain several pens with different colors of ink. Line and bar graphs, maps, and so on are drawn using several colors of ink. Weather maps, architectural designs, and engineering drawings are made more readable and useful by appropriate use of color.

Although computer graphics display screens are now commonplace, it was not always so. Computer graphics systems built during the 1950 s and 1960 s were quite expensive and required a large computer for their use. Thus, computer graphics was a tool of a few people in science and industry, and was an advanced course offered by computer and information science departments. But computer graphics was not readily available to the novice computer user or to pre-college students.

In recent years, progress in computer and television technology has made possible the use of television sets, or specially modified television sets, as inexpensive computer display units. Even the cheapest computer system can output simple line drawings. A microcomputer using a color television set can output graphs, charts, and simple pictures in full color.

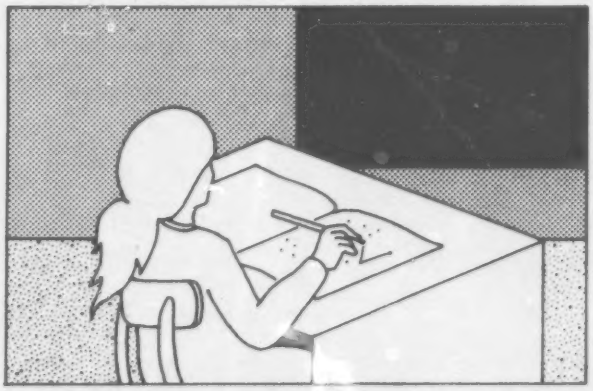
Moreover, it is now quite easy to make use of computer graphic facilities. The language Logo was specifically designed with graphics in mind. Thus, the young child learning Logo does so in a graphics mode. The child uses Logo to draw pictures on a TV screen or on paper. The programs that create these pictures are easily modified to correct bugs, and they are easily combined to create more intricate designs.

Computer graphics adds a new dimension to learning to program a computer. Rather than making it harder, computer graphics makes learning to program much easier (assuming that a graphics-oriented language such as Logo is available). A picture is much more concrete than a table of numbers. Students, especially young students, benefit from having such concrete output from a computer as they learn to program.

Applications

1. All students are familiar with connect-the-dots activities. A picture is formed by connecting the dots in numerical order. Students can create their own connect-the-dots activities and thereby learn some of the ideas underlying computer graphics.

   All that is needed is a picture, perhaps a simple line drawing, and a piece of tracing paper. The tracing paper is placed over the picture. Points are selected and indicated on the tracing paper. They are numbered in the order in which they are to be connected. It is assumed that straight lines will be used to connect successive points.



   The resulting picture can be made better by drawing in on the tracing paper some of the parts of the picture that are not easily formed by connecting dots. Because dots must be connected in a specified order, without raising the pencil off the paper, there are severe limitations to the quality of the picture.

   One can enhance the quality of the picture by allowing students to place a U or a D in front of some numbers. When the drawer comes to a number preceded by a U, s/he lifts the pencil off the paper. The pencil continues that way until one comes to a number preceded by a D. The pencil is then lowered down onto the paper.

2. Older children are familiar with plotting points on graph paper, or can easily learn to do this. This allows a more computer-like approach to the connect-the-dots activity. Now the idea is to give a list of coordinates of points that are to be connected. A small sample is given below.

   1. Raise pencil
   2. Move to (3,2)
   3. Lower pencil
   4. Move to (3,7)
   5. Move to (5,5)
   6. Move to (7,7)
   7. Move to (7,2)
   8. Raise pencil
   9. Stop

   As in Activity 1, students can create these sets of directions. One begins with a picture on graph paper (or place a very thin piece of graph paper over a picture). The coordinates to be connected are then read off the graph paper and written down. Students will quickly find the value of using brief abbreviations. For example, the previous set of directions can be abbreviated as follows.

   1. U
   2. (3,2)
   3. D
   4. (3,7)
   5. (5,5)
   6. (7,7)
   7. (7,2)
   8. U
   9. Stop

   In both Activity 1 and Activity 2 the idea is to have each student develop a set of directions. Students can then exchange papers and try out each other's set of directions. These sets of directions, especially the type developed for Activity 2, are much like computer programs.

3. An inexpensive microcomputer system is apt to use a television set as an output display device. In addition to displaying alphabetic and numeric characters, such a system can display very small rectangles of light. Thus, a picture made up of a large number of lighted rectangles can be created and displayed. Elementary and middle school students can learn similar ideas by working with graph paper, shading in the squares to create a pattern or picture. For example, students can spell out their names using letters that are made by shading small squares. Have

each student create a pattern or picture. Post some of them on the bulletin board.

4. A typewriter can be used to create computer-like graphics. In typewriter graphics one designs and types out a pattern or picture using just the typewriter character set. Varying shades of darkness can be formed by typing several different characters on top of each other.

   Bring a typewriter to your classroom. An old manual typewriter will suffice. Illustrate typewriter graphics to your class, and then have each student design a graphic on graph paper. Finally, have students type their graphics and display them on a bulletin board.

Artificial Intelligence

How "smart" can a computer be? Will computers eventually be smarter than people? Are the walking, talking, thinking robots of television and movies a possibility? Are there some problems that cannot be solved by computer because we cannot figure out how to write an appropriate program?

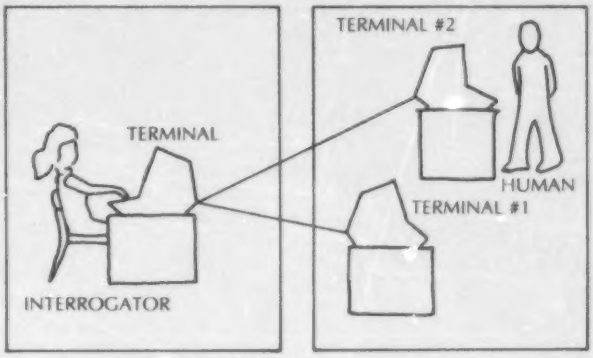
The branch of computer science that deals with these types of questions is called artificial intelligence (AI). When people are first introduced to the questions of AI, they automatically think in terms of human intelligence and start comparing computer capabilities with human capabilities.

While this is not inappropriate, the field of AI covers a broader range of topics. The capabilities of a simple pocket calculator can be considered to be an example of AI. Suppose, for example, that you had never heard of the idea of a calculator, and you were just learning how to do long division of decimal numbers. If someone showed you an $8 machine that could do such a problem in a few seconds you would be impressed. Indeed, you might think it was a pretty smart machine—that it was an intelligent machine. You might wonder why it was so important to learn to do long division using pencil and paper when the machine was so much better at it.

It is clear that computers can do many intelligent-like things. Computers can play games such as checkers and chess. Indeed, computer programs exist that can play checkers better than all but the very best human checkers players. Computers can do medical diagnostic work. They can analyze electrocardiograms. They can analyze the structures of chemical compounds, and they can solve very complicated mathematics problems. They can prove math theorems and can even appear to carry on a limited conversation in English.

The question of machine intelligence has been debated for many years and will continue to be a lively topic of conversation. In 1950, Alan Turing published an article that provides good insight into the problem of answering the question, "Can a computer think?" Turing was one of the earliest computer scientists and was also a brilliant mathematician. Some of his work done during the 1930s is still a standard part of the computer science curriculum at the college level.

Turing proposed a test for machine intelligence. The physical setup for the test is shown in the picture below.



The idea is that the interrogator can communicate with Terminal #1 and Terminal #2 only by terminal. Thus s/he can type and receive messages. By asking questions and carrying on a typed conversation with #1 and #2, the interrogator is to decide which is the computer and which is the human.

The human in Room B tries to help the interrogator by telling the truth, while the computer tries to fool the interrogator. That is, the computer tries to imitate human behavior. This is quite similar to the long-running television program "To Tell the Truth," in which two people pretend to be a third person. Turing's "imitation game" is a scientific experiment that can be run over and over again. Each time the computer and human in Room B are randomly assigned to the terminals. The same interrogator can thus run the experiment several times. One can also have a sequence of different interrogators.

If the computer can fool the interrogator about half the time, then Turing would say we have an intelligent computer. In 1950 he suggested that by the year 2000 we would achieve this. While considerable progress has occurred in developing a computer system to play the imitation game, we do not yet have a computer program that can consistently fool people.

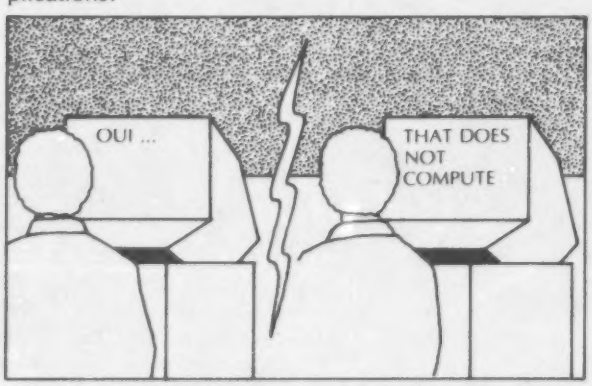
An early program, and still one of the best that can carry on a conversation with a human, is Joseph Weizenbaum's ELIZA. Weizenbaum is a computer science professor at the Massachusetts Institute of Technology. His program ELIZA, developed in the mid 1960s, is designed to interact with a person in a non-directive therapy mode. That is, the conversation is similar to that which might occur in the type of non-directive therapy pioneered by Carl Rogers.

Modern versions of the program are good enough so that they sometimes fool a person into believing s/he is actually conversing with a human, via terminal. But actually the program is quite simple and has very little in-

telligence. It merely picks out key words or phrases from what the human types and parrots them back with a slight change. We have a long way to go before a program is developed that will consistently pass the imitation game definition of artificial intelligence.

Part of the difficulty can be seen by examining progress in the language translation problem. At one time people thought it would be easy to develop a computer program to translate from one language to another. They tended to think of this as an automated superspeed dictionary. But language translation is not a word by word activity. Rather, it is an idea by idea activity. One must accept as input the material to be translated. One must then figure out its meaning—its basic ideas. Then one translates the ideas into another language. So far, computer scientists have made relatively little progress in developing programs that can extract the ideas and their meaning from written materials. They have not succeeded in solving the language translation problem and no solution is in sight.

We should not, however, downplay the success of artificial intelligence. Progress in certain aspects of chemistry, mathematics and medicine are very impressive. And many important problems have been solved. A relatively recent invention is a machine that can read a book. It employs an optical character reader to input printed material to a computer. The computer analyzes the words and produces spoken output of the printed material. Initially this system was designed for blind people. Can you think of other educational applications?



Problem solving is at the very heart of AI. We want to program computers to handle the unexpected—to be able to solve problems that are different from what the computer has previously encountered. It is easy to see how difficult this AI task is, since we do not yet have good insight into how to teach humans to solve problems. AI research into problem solving is leading to better insight into human intelligence. Also, it is leading to machines of ever increasing capability. Computers can do some things better than people. The number of things that computers can do as well or better than many people is steadily increasing. Needless to say, education at all levels should consider the progress that is occurring in AI.

Applications

1. The imitation game can be played with a child and an adult in Room B, with the interrogator trying to decide which is the child and which is the adult. No computer terminals are needed.

   Suppose, for example, that the adult decides to imitate being a child. The interrogator is placed in one room. Students are used to convey the messages. To begin, the interrogator might ask, "#1, are you an adult?" If the actual #1 is a child, the true response, "No, I am a child" will be returned. But if the actual #1 is an adult, she will send back a message such as, "Of course not, I am a child."

   Can you think of things that a child might know that an adult might not know? Remember, in this game one of the #1/#2's must tell the truth. The other is free to lie, in order to fool the interrogator. It's a fun game!

2. There are many machines that display some level of artificial intelligence. An elevator, for example, can store information, figure out the correct direction to go, and know when to open and shut its door. Discuss with your students the idea of smart machines. Then have students make a list of smart machines. Include in the list a brief description of what intelligent-like things each machine does.

   The activity can be extended by gathering pictures or having students draw pictures of the machines and prepare a wall display. The machines might be placed in order of increasing "intelligence."

3. Probably all of your students are familiar with the walking, talking, intelligent robots of movies, television and comic books. Have your students prepare a list of names and descriptions of these robots. What are their capabilities and limitations? Are they good or evil? What distinguishes them from humans? A major purpose of this activity is to help your students realize that these robots do not really exist.

Educational Implications

Now you understand that computer science is much more than just computer programming. Computer science provides new, exciting, and very useful ways of examining and attacking a wide variety of problems. These new methods supplement and sometimes may replace non-computerized methods.

The things that you have been learning about computers and computing are things that your students can learn. It is true that you are learning these things as an adult and that learning some aspects of computer science may strain your adult-level learning skills. But many secondary school students are currently studying these same things, and some are studying them at a considerably greater depth.

The International Council for Computers in Education has published a booklet, "An Introduction to Com-

puters and Computing" by Jean B. Rogers. The booklet is a detailed course outline for a secondary school course to be taught at the same level as a ninth or tenth grade introductory biology course. The course has a low mathematical prerequisite and is aimed at the general student population. The course includes major units on modeling and simulation, computer graphics, information retrieval, and artificial intelligence. Students spend four to six weeks on each of these topics. Eventually, much of this material may be integrated into the junior high school or middle school curriculum.

At the elementary and middle school levels, each part of the curriculum needs to be examined in light of current and potential computer capabilities. Three types of changes in the curriculum are necessary:

1. Modifications that lead to better preparing students to eventually study computers and computing in depth. For example, there could be increased emphasis upon modeling and simulation, and upon information storage and retrieval, from a non-computerized point of view.
2. Inclusion of computers and computing as part of each current curriculum topic where computers are a useful tool. The emphasis is still upon the non-computer curriculum topics, and computers are a supplement, or another way of attacking the problems of these areas.
3. Elimination of certain topics, or a great reduction in time spent on them. If a computer can solve a particular type of problem, what aspects of it should a student be able to do mentally or with pencil and paper? As computers become more and more available and capable this question grows in importance and difficulty.

Exercises

1. Examine textbooks for elementary and middle school students. Do any discuss modeling and simulation? Discuss your findings.
2. ERIC is a bibliographic educational information retrieval system. The system can also be used "by hand" since the indexes and abstracts are available on microfilm. Most major colleges and universities have access to the ERIC materials. Select a topic related to education. Learn to use the ERIC system well enough to do an adequate ERIC literature search by hand and/or by computer. Report on the results. Your report should include information on ease of learning to use the system, ease of using the system, satisfaction with the system, and overall results obtained. A computerized search can be done for about $15. Compare this with the cost of doing a similar search by hand.
3. An elementary school student can easily learn to use a Logo computer system to draw boxes, houses, flowers, robots, and other figures. With some instruction from peers or teachers, a student can add color and motion to these artistic creations. Use Logo yourself for a while. If possible, observe young students using Logo to help some students begin to learn the language. Then write a paper discussing potential roles of computers in the elementary school art education curriculum.
4. The computerized word processing systems now being sold for use in business offices often include a dictionary to check spelling. Suppose that elementary and middle school students were allowed to do much of their writing on a word processing system. The system would include a dictionary that, for example, would allow a student to spell a word phonetically and have the computer return a correct spelling and definition. Write a paper on educational implications of such a word processing system.

CHAPTER EIGHT

THE FUTURE

Writing about the future of computers in education is quite a bit like writing science fiction. Almost anything is possible, and the future is difficult to accurately predict. Therefore, this chapter consists of a sequence of speculations, and only time will tell if any will prove to be correct.

The predictions discussed in this chapter center around two main questions: What facilities will become available? How will the facilities be used?

The future of facilities—hardware, software and courseware—can be predicted with considerable confidence. Developments in this area are group efforts, led by companies driven by a profit motive. Education is only a small part of the hardware market, so hardware progress will be dominated by factors outside of education.

But the actual use of computers is more of an individual question. Even if high quality computer facilities are readily available, an individual teacher may decide not to use them. The teacher may lack the knowledge, confidence, or incentive to integrate computer usage into his/her classroom. Political factors, or unions, may play a dominant role. Thus, predictions in this area may not prove to be very accurate.

Hardware

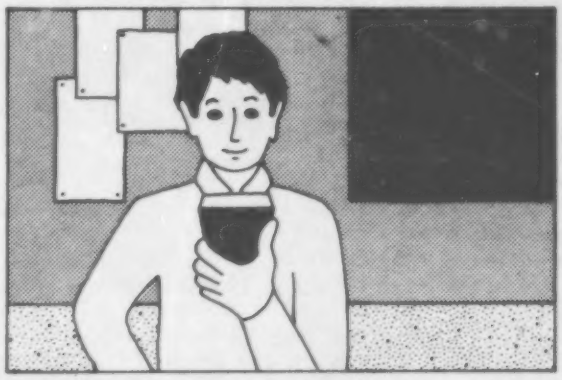
The nature of commercially available computer hardware can be accurately predicted up to five years in advance. That is because it takes about five years for products that are just now proving themselves in the research laboratory to become commercially available. Long term forecasts are more speculative, as they are based upon past and current trends.

The development of large scale integrated circuits continues at a rapid pace. Every few years we can expect a significant increase in chip density—the number of components on a single chip. Currently (1981), the 64K bit memory chip and the 16 bit CPU chip are just becoming available in large quantities. These will make possible microcomputers of substantially increased compute power (speed and primary storage). The more powerful machines will cost about the same as today's machines. The recently announced IBM microcomputer is a forerunner of such products. It has a 16 bit CPU, in contrast to the eight bit CPU's in the current Apple, Atari, PET and TRS-80 microcomputers.

By 1986 we can expect 256K bit memory chips to be available, and by 1991 we may well have a single chip containing both a million bits of primary storage and a 32 bit central processing unit. Some people predict that by the year 2001 we will have a 10 million bit chip!

The microcomputer of the early 1990s will be as powerful as many of the medium scale computers of the mid 1960s. Such mid 1960 machines cost a half million to a million dollars or more, and were adequate to serve the needs of an entire college or small university. When used in a timeshared mode, the medium scale computer of the mid 1960s could serve 30 or more simultaneous users.

Progress in chip technology will make possible handheld computers with far greater capability than the desktop microcomputers of today. If a million bit chip becomes readily available in the early 1990s, then by the year 2001 there will be inexpensive handheld games containing this chip. On gift giving occasions, people will receive toys containing more computer capacity than today's Apple, Atari, PET or TRS-80.



Bubble memory is quite likely to be an integral part of the next generation of microcomputers. Bubble memory is a type of secondary storage with no moving parts. Currently it is used in a few of the more expensive microcomputers, in a number of computer terminals, and in other places where a medium amount of very reliable secondary storage is needed. A 256K bit bubble memory chip has been commercially available for about two years, and a chip with four times this capacity is now in mass production. Very high volume, mass production of bubble memory is just beginning. It will cause a rapid drop in prices, making bubble memory competitive with floppy disks for many applications. Progress in developing larger bubble memory units will continue. By 1986 many microcomputers will

contain a magnetic bubble memory unit. This will increase the portability and ruggedness of microcomputers. By 1991 a microcomputer with a million bytes of bubble memory will be readily available.

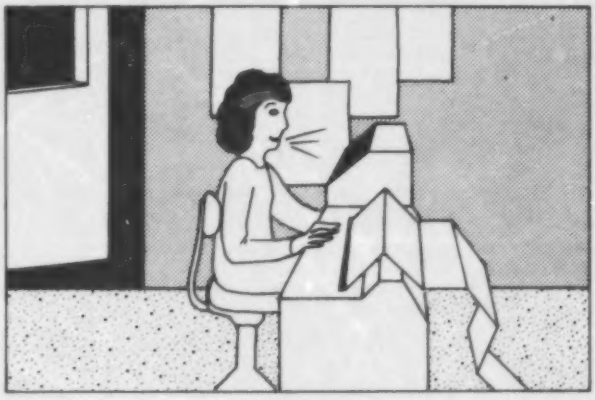
The microcomputers of 1986 will be designed to more easily interface with large computers and computerized data banks. Thus, it will be common to plug a bubble memory-equipped microcomputer into a larger system for the rapid transfer of a bubble memory of data. This may consist of lessons a student is to use that day or a large collection of data to be used in solving a problem. This type of setup will mean that many microcomputers will not need a floppy disk drive for secondary storage. Rather, they will serve as "intelligent" terminals, used both in a stand-alone mode and as part of a computer network.

Secondary storage for both small and large computer systems will grow in capacity. For microcomputers, the 5¼ inch floppy disk now storing 100,000 to 150,000 bytes will eventually have ten times this capacity. Mini hard disks with a storage capacity of five to ten million bytes are already available. They will decrease in price and increase in capacity over the next five to ten years. By 1986 many microcomputer systems will have a half million byte floppy disk or a 10 to 20 million byte hard disk for secondary storage.

Relatively few microcomputer users need exclusive access to a 10 to 20 million byte data base. A network of microcomputers sharing a hard disk and a printer has been commercially available for several years. Such networks will become increasingly common in schools. They are especially useful when one has a cluster of fifteen or more microcomputers in a single classroom.

The technology for high quality, low cost, audio output—both voice and music—is already available. Software for easy use is being developed. By 1986 good quality voice and music output will be available on many microcomputer systems. More computers will give spoken directions to their human users. This is a relatively straightforward extension of Texas Instrument's Speak and Spell and other microprocessor-based "talking" devices of 1978-1980. When purchased in large quantity, the current wholesale price of a high quality voice synthesis chip is less than $10.

Progress in voice input will continue. If one is willing to accept a small vocabulary and relatively high error rate (perhaps several percent), then inexpensive voice input is already available. A larger vocabulary and a lower error rate require more compute power (CPU speed and primary storage capacity). Since rapid progress is occurring in both of these areas, we can expect voice input to be a readily available option by 1986. Voice recognition of "connected speech" (that is, normal spoken phrases and sentences) is still a major research project. By 1986 significant progress will have occurred and by 1991 such systems will have some commercial applications. It will be interesting to see if the voicewriter of science fiction books is available at a reasonable price by the year 2001. Clearly this is a substantial challenge to researchers. Perhaps typing classes in the year 2001 will be teaching clear pronunciation instead of touch typing!



We can expect continued progress in the development of videodisk systems interfaced with microcomputers. Ten years from now many home entertainment centers will include a general purpose microcomputer, videodisk, video recorder, and television display. This may be tied into a large centrally located computer via telephone line or TV cable. Perhaps ten million homes in the United States will have such a system. A videodisk may contain entertainment material. Alternatively, a single videodisk can store the equivalent of several hundred textbooks or reference books.

Work is currently in progress to develop an inexpensive unit to record videodisks in one's home or office. By twenty years from now such units will quite likely be common in business and industry, and may be common in people's homes. Since a single videodisk can store hundreds of millions of characters of data, this will contribute significantly to the computerization of libraries and other large collections of data.

Computerized videodisk home entertainment centers will be backed up by high quality software, both recreational and educational. The educational software will be competitive with "conventional" instructional delivery systems in a wide variety of subjects. The potential impact upon elementary and middle schools is great. This issue will be briefly discussed later in the chapter.

Very rapid progress will occur in telecommunication systems, allowing home and school computer systems to connect to large data banks. Already fiber optics is a well developed technology. A glass fiber as thin as a human hair can carry the equivalent of 2,500 simultaneous telephone conversations or 20 television signals. This technology will begin to replace the copper telephone wires and the copper television cables coming into people's homes. It will make possible relatively inexpensive "picture phones" and interactive TV. It will contribute substantially to the development of multinational computer networks of computer systems located throughout the world.

Communication satellites will become an increasing part of our telecommunications system. A key point is that the actual cost of relaying a signal via satellite is independent of the distance involved. If one is communicating via a satellite, the actual cost to the satellite owner is the same for a Los Angeles to San Francisco call as it is for a Los Angeles to New York call.

So far in this section we have emphasized microcomputers. But the same rapid progress will occur in large scale computers. For such machines we may expect a gain in total speed by a factor of ten during the next 5-10 years, and another factor of ten during the subsequent decade. Large scale computers with a speed of a billion instructions per second will be common by the year 2001, and the very fastest machines will be 50-100 times this fast.

Currently, a large magnetic disk pack (a single disk storage unit) can store 300 million to 1,000 million characters; storage devices based upon laser technology store several hundred billion characters. Continued progress will occur both in magnetic storage technology and in laser-based storage technology. The result will be computerized access to data banks containing trillions of characters. Recall that a very thick novel is about a million characters, and that a trillion is a million million. Thus, we will have storage devices that will fit into a small room, perhaps even sit on a desk, that can store the entire contents of the Library of Congress.

Very large storage devices, coupled with increasingly capable telecommunication systems, will change the nature of research libraries and the publication of research results. Articles will be "published" by entering them into a data bank accessible by computer terminals from throughout the world. Researchers and research libraries will pay to access these data banks, and will have access to published research the moment it is put into the data bank.

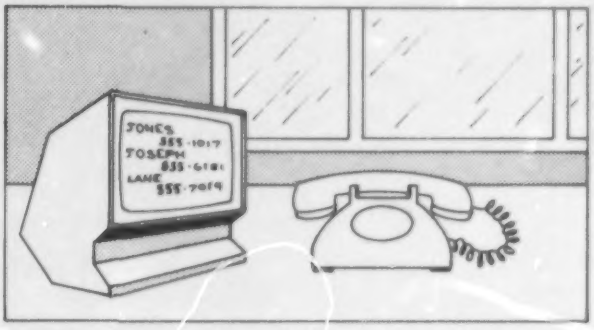
Software

Computer scientists frequently divide software into two categories. Systems software consists of language translators (BASIC, COBOL, Logo, Pascal, PILOT, etc.), operating systems, data base management systems, general purpose information retrieval systems, and so on. Each is designed to provide a general set of capabilities to the user, rather than to solve a specific type of problem.

Applications software consists of programs designed to solve specific types of problems or to accomplish specific tasks. Examples include programs to process a payroll, solve an equation, perform a statistical correlation, drill a student on spelling words, or quiz a student on number facts. CAL programs are applications software designed to help students learn. CAL software, combined with teacher resource guides and other materials to aid students and teachers, is called CAL courseware, or simply courseware.

There is no fine dividing line between systems and applications software, and the distinction seems to be blurring with time. A tremendous amount of research and development is currently being expended on systems software. The goal is to provide better tools for both programmers and to people who have not received extensive training in the use of computers. There seems no inherent reason why a person needs to be a computer expert to use a query language to retrieve information from a computerized library. Similarly, a business person should be able to use a word processing system and an electronic mail system. We can expect substantial progress in all of these areas. Many secretaries in the year 1986 will be adept at word processing, use of electronic mail, and computerized storage and retrieval of data.

Retrieval of telephone directory information provides an example of expected progress. Already telephone operators use computerized directories, with operators located in a single city serving a whole state or larger region. In both France and the United States there are ongoing experiments, with computer terminals in people's homes being used to access telephone directory information. In ten years, home computerized telephone directory access will be fairly common. The French government has plans to replace all home telephone books with such terminals; mass production of the terminals has already begun. These same terminals can give access to data banks containing other information and could be the basis of a nationwide electronic mail system.



Other types of computer-based information retrieval in people's homes are already commercially available. Several approaches are being used. One method uses standard broadcast TV signals. A TV picture consists of a large number of lines; the signal broadcast by a TV station contains a few more lines than are needed to generate a picture on the TV screen. These additional lines can be coded as characters of information, received and decoded by a special device added to the TV set, and displayed as pages of printing. The TV station can maintain an up-to-date data bank of pages of information. These pages might be sports scores, stock market quotations, weather forecasts, world news, and so on. In this way the TV user has access to hundreds of pages of very current information.

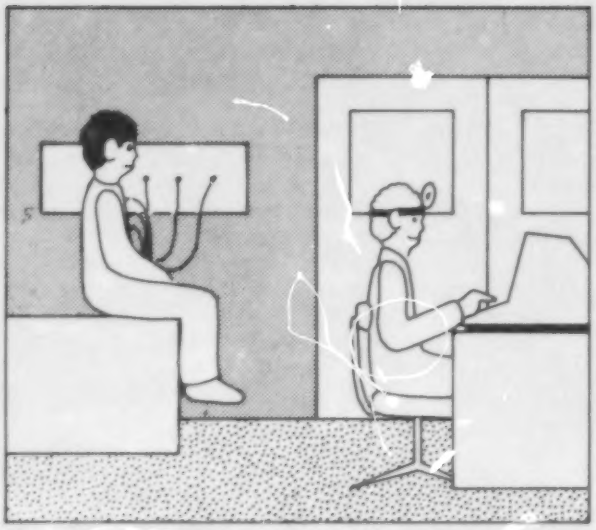
A second approach is via telephone lines to a very large data bank, with output being displayed on one's home TV set. A third approach is via cable TV. All cable TV systems currently being installed in the United States have two-way capability. Thus, they are designed to permit transmission of signals from a TV station into a home, and also transmission of a signal from a home to a centrally located computer. This means one can have an interactive information retrieval system not making use of the telephone. All three of the systems we have mentioned are currently in use, and it is not yet clear which, if any, will dominate in the long run. What is clear is that computerized IR in people's homes will become increasingly common.

Another type of progress is a natural outgrowth of increased hardware capabilities of microcomputers. Already microcomputers can use languages such as COBOL, Logo, FORTRAN, Pascal and PILOT that were formerly available only on large scale computers. Indeed, the microcomputer of the future will not be "micro" in capability, and a name change would be helpful. The trend of rapidly increasing capability of easily portable computers will continue, so that microcomputer users will have access to most software now found on large computer systems.

The next fifteen years will witness the birth of some new programming languages, profoundly different from the high level languages currently available. They will be designed to help people who know little about computers or programming to clearly specify tasks a computer is to perform and problems a computer is to solve. These languages will be much easier to learn and to remember than Pascal or similar currently used languages.

These new languages will be highly interactive, designed to help the user describe a problem situation and to work through possible methods of attacking the problem. Computer systems using these new languages will be quite intelligent, making suggestions and asking questions to resolve ambiguous specifications by the user. Initially such problem-solving systems will require very large computer systems, but twenty years from now such aids to problem solving will begin to become available at a reasonable cost to large numbers of people.

Along with the new languages will come a proliferation of knowledge-based computer systems. These are interactive computer systems that contain a substantial amount of knowledge about the problems in a specific discipline, and how to solve the problems. A math system will "know" how to solve problems from algebra, calculus, linear algebra, logic, and so on. A medical system will be a good diagnostician, interacting with a doctor or patient. A chemistry system will know how to solve many of the problems covered in current high school and college chemistry courses. Students at the secondary school and college levels will learn to use these systems; the impact upon the current, conventional curriculum will be substantial.



Applications software, especially CAL courseware, will make rapid strides. Right now the production of CAL courseware is largely a cottage industry. It is only in the last year or so that major textbook publishing companies have begun to gear up to produce computerized supplements to their materials. As computers become more readily available to students, the publisher who takes advantage of this instructional medium will gain a sales advantage. Other publishers will then jump on the bandwagon, and computerized supplements will become available for most of the nationally distributed textbook series. Over the next ten years, however, textbooks will likely continue their dominant role as a medium for the dissemination of information.

There are not yet enough microcomputers in schools or homes to make courseware development a profitable venture for very many people. It is estimated that in the fall of 1981 there were about 100,000 microcomputers being used for educational purposes in the United States. This is changing rapidly, however, with sales of microcomputers continuing to increase on a year to year basis. Educational sales for the year 1981 are estimated at 70,000 machines. Ten years from now there will probably be many millions of microcomputers capable of using CAL materials. A mass market for CAL courseware will exist, and large quantities of materials will be available.

Quality control will continue to be a problem, however. The development of good CAL material and supporting aids for teachers and students is both difficult and expensive. There is a severe shortage of people who are qualified to develop good CAL courseware. Moreover, there are few educational programs that can train such people, and few students enrolled in these programs. Ten years from now educators will still bemoan the lack of adequate, good quality courseware even though very substantial progress will have occurred.

Looking still further ahead, perhaps ten to twenty years from now, the available combination of hardware and courseware will begin to profoundly impact education. CAL will not only supplement standard textbooks, but will begin to replace many of them. Good quality CAL to help students learn reading and arithmetic will run on a home entertainment center, and some of it will be suited to the needs of preschool children. Thus, increasing numbers of students will learn to read and do simple arithmetic before they start school. CAL materials will be commercially available for almost all courses at the precollege level, and for undergraduate college courses. Adult education, vocational education, hobby education, and so on will begin to occur via computer.

Alfred Bork, one of the leading experts on computer assisted learning in the United States, has predicted that by the year 2000 more than half of all education in the United States will be via CAL. That is a bold prediction, and certainly provides food for thought. Inherent to Bork's prediction is the suggestion of a massive change in education. Currently in the United States most education is formal, via schools grades K-12, community and junior colleges, colleges and universities, trade schools, etc. The learner and the sources of learning are physically brought together in a classroom setting. Much of the responsibility for learning is placed on the instructor.

One model of massive change is the Open University in England. Tens of thousands of people are working towards college degrees using courses presented via public television. This has proven to be a highly cost effective method of making college education available, even though it may cost a million dollars to develop a single course. Funds for the development of these courses come mainly from the government, and a single course is used by tens of thousands of students.

The computerized home entertainment center adds a new dimension to the Open University concept. It may well cost several million dollars to develop a high quality CAL course. But eventually the potential audience will be sufficiently large to justify such an expense. When this movement becomes well established it will change the face of education. The topics that can be effectively taught via CAL will indeed by taught using this medium. The equipment will be located in homes, libraries, science museums, places of work, and schools. Children will have access to high quality CAL long before they are old enough to begin attending school.

Another model for educational change is a massive expansion of private education or in-the-home education by parents. The growing dissatisfaction with public education may lead to voter approval of voucher systems, with tax money going to parents who then select the schools their children will attend. Currently about one-sixth of all school-age children in the United States attend private schools. Good quality CAL will help small private schools to offer a strong program in the "basics" while keeping their costs down. As Logo and similar CAL systems become cheaper and readily available, such computer usage may become a standard part of the private school curriculum.

The computer impact upon formal education at the elementary and middle school levels will be large, but twenty years from now these schools will likely still look much like they do today. Students will still come together in schools and will still have human beings as teachers. More teacher time will be spent on developing the interpersonal, human-to-human skills. Less teacher time will be spent on record keeping, paper grading, providing individual feedback to students on common subject matter being learned, and so on. The computer will become the dominant instructional tool, performing the functions of direct instruction, testing, record keeping, and guidance of students through material to be learned.

Business and Industry

The hardware emphasis in this book has been on microcomputers, since they most directly impact precollege education. But large scale computer systems are still in considerable demand and their use is expanding rapidly. Right now each of the manufacturers of large scale computers has a large backlog of orders and some are hard pressed to meet the demand. The amount of compute power represented by this backlog of orders exceeds the total currently installed capacity of all large scale computers in the world. Large computer manufacturers, such as IBM, are hastily adding to their production capacity in order to meet this growing need.

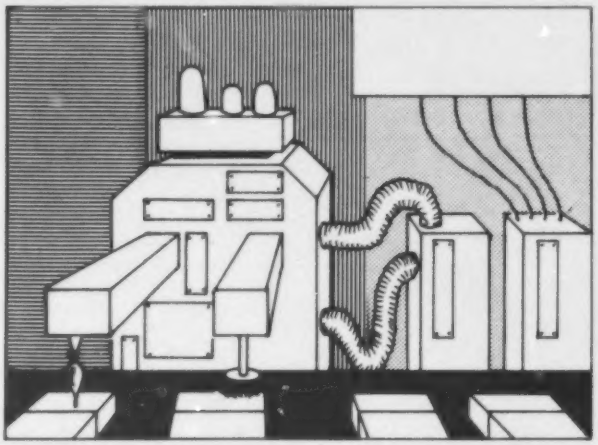
The key to expanded use of computers in business and industry is increased worker productivity. Take a simple example. A secretary can be trained and skilled at very neatly writing out letters in longhand. Give the secretary a manual typewriter and productivity may double. Give the secretary a self-correcting electric typewriter and another significant gain in productivity results.

Now supply the secretary with a computerized word processing system, access to a computerized dictionary and other data banks of information, and access to an electronic mail system. Productivity again takes a significant jump. This is just beginning to occur in the

typical business office. Ten years from now word processing and related applications of computers in the secretarial office will be commonplace.

You are undoubtedly aware of the massive change in farm employment that has occurred in the United States over the past two hundred years. At one time nearly ninety percent of the population worked on farms to produce the food needed in this country. Gradual but continual increases in farm worker productivity have led to the situation where now less than four percent of our labor force works on farms, and this country is a massive exporter of foods.

We are just at the beginning of a new industrial revolution—the robotization of industry. An industrial robot is a computerized machine that can weld, screw nuts on bolts, fit parts together, run a spray painter, and so on. Typically an industrial robot costs $50,000 to $100,000 or more, and performs a task similar to a human worker. But the robot tends to be faster, more reliable, and able to work several shifts per day. (Routine maintenance, plus repair when the robot breaks down, are necessary, however.) The price of industrial robots will decrease as they begin to be mass produced.



Right now there are only a few thousand industrial robots in use in the United States, and perhaps twice that many in use in Japan. Japan and the United States are leading the world in the development of these versatile, computerized aids to increased manufacturing productivity. The currently available models are good—well worth the money; demand is large and is growing rapidly. Over the next two decades there will be a massive influx of robots into industrial production. Millions of human workers will be displaced and/or will need to be retrained for such tasks as robot maintenance and repair. Of course, we will also have robots that can maintain and repair other robots. One of Japan's leading robot manufacturing companies is currently building a large factory and equipping it with robots that will build robots. They expect to capture a significant percentage of the rapidly expanding world market for these aids to industrial production.

In the past, college educated people have tended to feel that automation was something to be feared by assembly line workers, but would not directly affect people with college-produced skills. The computer has changed this. A computerized autopilot can take off, fly, and land a plane as well as a very skilled pilot. A computerized typesetting system displaces skilled linotype operators. A computerized graphics system displaces hundreds of cartoon artists used to produce children's television cartoon programs.

And we must not forget education! Computers can increase teacher productivity; this could lead to a decrease in the total number of employed teachers.

Education

By now you should be convinced that computers have the potential to massively impact education at all levels. CAL can help students to learn more, better, and faster. CAL can shift the responsibility for learning more towards the learner and can increase the amount of learning going on in out-of-school situations. Much instruction that currently requires the presence of a teacher can occur in the home, library, or place of work, via CAL.

But CAL is only part of computers in education. Students need to learn to use and to live with computers, so instruction about computers needs to be integrated into the curriculum. Computers can solve, or can assist in solving, many of the problems students currently study. The curriculum must change to reflect and to take advantage of this.

Most educators do not yet comprehend the magnitude of the tasks mentioned in the previous paragraph. The entire curriculum needs to be carefully reviewed. All curriculum materials need to be examined, and most need to be revised. All teachers need some retraining, and most need a substantial amount. If you have carefully studied this book up to this point, then you have put in a significant amount of time and effort. Let us suppose that you have also worked on computer programming an equal amount, and have developed modest skills in programming.

You should be proud of yourself, because your knowledge of computers places you in the upper ten percent of all elementary and middle school teachers. But—you should also realize that even your knowledge is not adequate to the curriculum revision tasks we have described! Would you have teachers who had a single college level course in mathematics (and no pre-college work in this area) revise the math curriculum? Would you have teachers who had just learned to read via a single college level reading course be responsible for establishing the elementary school reading curriculum? Would you have teachers who had never even heard music before taking a single college level music course be responsible for developing and implementing the music curriculum?

Over the next two decades, educators will come to realize that computers are equivalent to reading and writing, or to arithmetic, as part of the "basics" of education. Students who receive an education fully integrating computers as a topic of instruction, an aid to problem solving, and as an aid to learning, will have a tremendous advantage over those who do not. Educational leaders and the general public will come to expect that schools will provide this education. Students will grow up with computers, using them as everyday tools.

Not all teachers will be able to adjust to the influx of computers into education. If you have learned the material in this book and have learned to program a computer, then you have demonstrated the capability of being a leader in this educational change. Take the knowledge you have, and use it! But be aware that there is much that you do not yet know. By all means, continue your informal education by reading, experimenting with your students, talking to other teachers about computers, and so on. If at all possible, continue your formal computer education. Get your school district to sponsor a workshop or an inservice course. Take an evening course at a college or university. Attend summer school. As you learn, use your knowledge. Integrate it into your everyday school activities, and pass it on to your students. The future of computers in education depends to a large extent upon YOU, and others like you. The future is bright if you and other teachers continue to learn!

APPENDIX B

PRE-COLLEGE COMPUTER LITERACY: A Personal Computing Approach

ABSTRACT

It is generally agreed that all students should become computer literate, but no definition of computer literacy has gained widespread acceptance. This booklet defines computer literacy in a manner that can guide educators as they work to implement universal computer literacy through precollege education.

This booklet is an updated and expanded version of a paper, "Personal Computing for Elementary and Secondary School Students," prepared by David Moursund for a computer literacy conference held in December 1980 in Reston, Virginia. The conference was organized by the Human Resources Research Organization and the Minnesota Educational Computing Consortium. The purpose of the conference was to help participants gain an increased understanding of the meaning of computer literacy and what can be done to help students become computer literate.

This booklet is intended for curriculum specialists, elementary and secondary school teachers, media specialists, teachers of teachers and others concerned with curriculum in precollege education. It defines and discusses computer literacy for elementary and secondary school students. The approach is via an analysis of personal computing and the aspects of computers that can have a direct impact on students. Students can be personally involved with computers through computer assisted learning, computer assisted problem solving, the study of computer and information science and through the use of computers for entertainment. Students can learn how computers are affecting the world of business, government and industry—and thus, how computers will be part of their future. Each of these aspects of personal computing contributes to the definition of a set of goals for computer literacy in elementary and secondary schools. The resulting overall goal is for a working knowledge of computers—that is, knowledge that facilitates the everyday use of computers by students. This knowledge lays a firm foundation for future learning about computers and for coping with the inevitable changes that will occur in this technology.

PRECOLLEGE COMPUTER LITERACY A PERSONAL COMPUTING APPROACH

PERSONAL COMPUTING (noun phrase) Easy to use, readily available, inexpensive access to electronic digital computers for personal use. The term gained prominence in the late 1970s with the advent and rapid proliferation of microcomputers. Since then prices have continued to decrease, while the quality and capability of microcomputers have increased substantially. The first commercially available handheld computer, introduced in 1980, and a variety of briefcase-sized computers introduced since then have lent credence to the idea of people having easy and nearly unlimited access to computers for everyday, personal use.

HISTORICAL OVERVIEW (TECHNICAL)

The history of computers can be viewed in terms of progress toward making computer systems more readily available and easier to use. The first stage was to make computers available—to invent the fundamental ideas and to build the first machines. During the late 1930s and early 1940s, substantial progress occurred in England, Germany and the United States. The first general-purpose electronic digital computer built in the United States was the ENIAC, which became operational in December, 1945.

The ENIAC and other early vacuum tube computers were difficult to use. The development of assembly languages and assemblers helped. But still, each computer required a team of electrical engineers and technicians to insure operation, and the machines were not very reliable. Computer memories were quite small and internal instruction sets (machine languages) were restrictive. The process of preparing programs and getting them into machine usable form was exacting and time-consuming.

By 1951, however, many of the initial problems had been overcome and the UNIVAC I, the first commercially produced computer, became available. Over one hundred of these machines were eventually produced and sold, evidence of a rapidly expanding market for computers. However, the UNIVAC I and other computers of the 1950s used vacuum tubes. Maintenance and reliability remained major problems, along with

the size of both primary and secondary storage and a shortage of good software and programmers.

During the 1950s, high level programming languages such as FORTRAN, COBOL and ALGOL were defined and implemented. Transistors became readily available, as did primary memory which made use of tiny iron cores. The second generation of computers that emerged in the early 1960s represented tremendous progress toward making computers more readily available, reliable and convenient to use. Some of these machines remained in service for 15-20 years.

Rapid technological progress in both hardware and software continued. Timeshared computing, especially interactive BASIC, provided a new standard of personalization in computing. Even more importantly, however, the 1960s saw the development of a process to manufacture a single (integrated) circuit containing a number of interconnected transistors and related components. Component density rapidly mounted from tens to hundreds to thousands on a single silicon chip. The cost per active component dropped rapidly, and reliability increased.

Large scale integrated circuitry helped define the term "third generation" of computers in the mid 1960s. Tens of thousands of computers were manufactured and installed. Minicomputers were priced so that an individual researcher or research project could own one. Computers became an everyday tool for hundreds of thousands of people. Since then, progress has been relatively smooth and continual. Thus, there is no agreed on definition of fourth or fifth generation computers.

The era of personal computing began with the introduction of microcomputers in the latter half of the 1970s. Suddenly it became possible for an individual to own a computer for use at home, in the office or in the classroom. The cost of teaching using computers rapidly dropped by almost an order of magnitude. Students, even in the elementary grades, could have hands-on experience with computers.

In 1980, a battery powered computer, the size of a handheld calculator and programmable in BASIC, became commercially available at a retail price of about $250. By 1981 this machine was being discounted to $200. In 1980 and 1981 several companies introduced easily portable briefcase-sized computers. Their advertising campaigns stressed the idea that these computers could be taken on business trips, thus having them available for use at all times.

In 1981, Hewlett-Packard announced a 450,000 transistor chip. This single chip contains more circuitry than many of the large scale computers of the early 1960s. In 1982, Casio began selling an ordinary-sized wristwatch which contains a 1,711 word Spanish-to-English and English-to-Spanish dictionary as well as the ordinary wristwatch functions. December of 1982 saw the sale of many battery-powered, handheld electronic games as well as tens of thousands of computers. Estimates are that two million personal computers were purchased for home use in the United States in 1982, and that four million more will be purchased in 1983.

These advances have opened up the possibility of personal computing for the general population. Computer manufacturers could produce a microcomputer for every home, office and student desk at school. These individual units could be connected to each other and to larger computers through our telephone systems and our cable television systems. Not only is this possible, but it is quite likely to occur over the next 10-20 years.

HISTORICAL OVERVIEW (EDUCATIONAL)

A number of the early computers were built at university campuses and were immediately used for both research and instruction. But first generation computers were relatively expensive, so their use in education was limited.

The 1960s saw a rapid proliferation of computers in education, especially in colleges and universities. Hundreds of computer science and data processing degree programs were started. Computers were easy enough to use so that college undergraduates could take a few computer courses and emerge from college with well paying jobs as computer programmers or system analysts.

By the early 1960s a few secondary schools had computer access and a few teacher training opportunitites were available via evening, weekend or summer courses. Later in that decade and in the first half of the 1970s, the National Science Foundation funded a number of inservice and summer institute computer courses for secondary school teachers. Most participants were mathematics or science teachers. Typical institute courses covered programming in FORTRAN or in BASIC, and some computer-oriented mathematics. The impact upon precollege education at that time was minimal, since few secondary schools had more than a modest computer facility available. In recent years, however, many of these early computer institute participants have emerged as computer education leaders in their schools and school districts.

Since the late 1950s, people have worked to develop computer assisted learning systems. Stanford University professor Patrick Suppes received substantial federal funding during the mid 1960s to develop and test drill and practice materials in elementary school arithmetic and language arts.[3] Many of today's drill and practice materials can be traced back to the pioneering efforts of Suppes' group. Computer assisted learning has become a major application of computers in precollege education.

The Colorado Project[1] was a leading example of early efforts to make significant use of computers in high school mathematics. This second year high school algebra and trigonometry course was developed in the late 1960s. Students studied BASIC during the first few weeks of the course and throughout the remainder of the course were expected to write short programs as part of their efforts to learn algebra and trigonometry.

The popularity of this course probably peaked in the mid 1970s and has since declined. For example, at one time nearly 10% of the high schools in Oregon offered at least one section of the course. But overall, few teachers felt qualified to teach the course, and the inclusion of computer programming into an already crowded course added to the difficulty of teaching it.

Despite such curriculum problems, instructional use of computers increased steadily through the mid 1970s and then accelerated as microcomputers became available. By the end of 1982 there were an estimated 200,000 microcomputers in use in precollege education in the United States—about one for every 250 students.

These microcomputers were not evenly distributed. In Eugene, Oregon (the author's home town) for example, there was one microcomputer for every 90 students. Every junior high school and high school in this town of 100,000 population offered computer programming courses and made some use of computers in non-computer courses. Several elementary schools were experimenting with computers, using both the BASIC and Logo languages.

An added impetus for computer science instruction in high school is provided by the Advanced Placement Test (students can earn up to a year of college computer science credit) that will become available in the spring of 1984. Roughly one-third of United States high schools offer an advanced placement-oriented calculus class. Most of these schools and many others may eventually want to prepare students for the advanced placement computer science course.

HISTORICAL OVERVIEW (COMPUTER LITERACY)

The idea of computer literacy for the general student population probably first emerged in the late 1960s. Leaders in the field of computer and information science began to suggest that all people needed to know something about computers. The Conference Board of the Mathematical Sciences recommended universal computer literacy in its 1972 report,[2] suggesting that this could be achieved via a junior high school computer literacy course. Although its request to the National Science Foundation for funding to develop such a course was denied, many individual teachers began to offer computer literacy units or entire courses, and authors began to develop materials useful at the elementary and secondary school levels.

The meaning of computer literacy has never been particularly clear, and it seems to have changed over time. Initially, computer literacy usually was taken to mean a level of understanding which enabled students to talk about computers but which involved little or no experience in working with computers. (This is now called computer awareness.) Students were exposed to movies and talks about computers, allowed to handle a punched card, discussed ways that computers were used in business, government and science, and perhaps toured a computing center. Little or nothing about this was personally relevant to most students, and being aware of computers had little impact upon them.

The growth of computer assisted learning added a new dimension.[3] The computer could teach the student. Certainly this had a direct personal impact upon students. Initial studies suggested that in some academic areas many students learned as well or even better from computers as from conventional modes of instruction. Computer assisted learning required little specific knowledge about computers on the part of either students or teachers, and computer hardware could be mass produced—a few people predicted that conventional formal education was doomed!

This situation prompted Art Luehrmann and others to raise an important issue in the early 1970s: what are appropriate uses of computers in education?[3] Should the computer teach the student—or vice versa? Or, as Tom Dwyer put it, should the student be the passenger or the pilot?[3]

The basic issues involve what students should learn about computers and the ways they should learn about or use them. When a computer acts upon a student in a computer assisted learning mode, the student need not learn much about computers. But when a student acts upon a computer, developing programs and solving a variety of problems, more knowledge of computers is required—on the part of both student and teacher.

The issues have been sharpened through the work of Seymour Papert. For more than a decade he has been developing the Logo language, turtle geometry, and ideas on using computers with elementary school children. Papert advocates immersing children in a problem-rich environment, and he has shown how computers can help provide this environment.[4] His work suggests that even very young children can become adept at using a computer as an exploratory tool and can learn key ideas such as top down analysis, debugging and subroutine. Papert questions whether our current educational system can cope with the changes he is advocating.

The issues raised by Dwyer, Luehrmann, Papert and others have not been resolved, in part because computer literacy is not accepted as an important goal by the majority of parents, educators or students. Even now the school that can provide one microcomputer or timeshared terminal per 25 students is rare. Rarer still is the school that has even one teacher with a knowledge of computers in education equivalent to a strong bachelor's degree in this field. Contrast this with almost every other academic area taught in secondary schools, where a bachelor's degree or an even higher level of teacher preparation is common.

However, both of these situations are changing—they could change quite rapidly if our society, working through its school system, decided that it was important to have it happen. The increasing personal access to computers may provoke that decision. The remainder of this booklet discusses some aspects of personal

computing and how they help to define some specific goals for computer literacy.

PERSONAL COMPUTING: A FORMAT FOR EXPLORATION

Students of all ages can learn to use a computer at a level that is meaningful to them and makes a difference in their lives. Personal computing for students can be divided into several categories. The following uses of computers are neither disjointed nor all-inclusive, but will serve to guide our exploration of the concept of computer literacy. Computers can be personally useful to students as:

I. A General Aid to Learning.
II. An Aid to Problem Solving.
III. An Object of Learning in Itself: The Discipline of Computer and Information Science.
IV. Entertainment.
V. A Part of Their Future.

Each of these will be discussed, along with how each contributes to a definition of computer literacy. The discussion will center around students and their everyday, in-school activities. Thus, the goals for computer literacy that emerge will tend to be student-oriented and relevant to students. Moreover, these goals will be flexible and easily modified as changes occur in computer capability and availability as well as in the curriculum.

I. A General Aid to Learning

Computer Assisted Learning, Tutor Mode

Computer assisted learning (CAL), the use of computers as an aid to learning, can be divided into two major parts. In one part, frequently called computer assisted instruction (CAI), the computer acts upon the student. Whether the mode is drill and practice, tutorial or simulation, the computer has the knowledge and it is the student who is to acquire the knowledge. Following Robert Taylor's ideas,[3] we will call this **tutor mode CAL**.

A second form of CAL puts the student in charge—the student acts on a computer as an aid to learning. Learning environments created using a Logo language-based computer system provide a good example.[4] After a few minutes of instruction, even an elementary school student can learn enough Logo programming to begin encountering interesting and challenging geometry problems. Immediately the emphasis then switches from learning Logo to problem solving in the domain of geometry. We will call this **tutee mode CAL.**[3] It will be discussed later in this section.

In essence, tutor mode CAL is a computer simulation of certain aspects of teaching/learning processes. The field is more than twenty years old now and is slowly maturing. Initially much tutor mode CAL material was quite poor, and even today this remains a major problem. But, like any computer simulation, tutor mode CAL quality can be improved by continued work on the underlying theory, the software, the hardware and the other supporting materials. There are now some quite good tutor mode CAL materials, with strong evidence that many students learn better and/or faster using these materials. Moreover, tutor mode CAL is an excellent educational research tool, contributing significantly to an understanding of what students learn and what helps them to learn. Good tutor mode CAL embodies what is known about learning theory and makes explicit the model(s) of instruction being used.

The explicit implementation of learning theories in tutor mode CAL is a key idea. All teachers have some insight into a variety of learning theories and realize that not all students learn in the same way. Significant progress in learning theory has occurred during the past twenty years. It is nearly impossible for a classroom teacher to keep up with this progress and to make use of the more relevant ideas in his/her teaching. However, modern theories of learning can be used in the design and implementation of tutor mode CAL materials. An individual or team of tutor mode CAL developers can spend the necessary time to study current learning theory research and to implement ideas that will help students learn more, better and faster. Tutor mode CAL can provide an individualization of instruction that cannot be matched by a teacher who must deal with large numbers of students.

Another important idea that can be made explicit in education is learning about learning. A student needs information on how, and under what conditions, s/he can learn best. That is, a student needs to learn about learning. The computer provides a good motivation and vehicle for specific instruction on learning and on learning to learn. What does it mean to "know" a particular topic? Are some methods of studying more productive than others? Is there one best method for studying all subjects? Since computers can help most students to learn faster, most students can benefit from learning the capabilities of tutor mode CAL and from having access to tutor mode CAL. Any student can learn to use tutor mode CAL and all can learn how (for them personally) CAL compares with other aids to learning.

Ideally, a computer literate student has experienced tutor mode CAL in a variety of disciplines and has developed insight into its value relative to other modes of instruction/learning. The student has used drill and practice, tutorials and simulations in art, music, math, science, language arts, social studies and so on. This has occurred at each grade level. The student has studied and thought about what it means to learn and has specifically studied various modes of instruction/learning. The student understands what best fits his/her needs in a wide variety of situations.

Notice that this aspect of computer literacy is sensitive to changes in computer technology and to changes in tutor mode CAL quality and availability. We need to acquaint students with the best that is currently available and help them to understand that this "best" is

rapidly changing. We should also stress that tutor mode CAL can occur in many settings outside the classroom and can therefore play a useful role in lifelong education.

This approach to computer literacy can begin at the elementary school level and can continue throughout a student's education. It has the potential to help revolutionize education. The responsibility for learning can be placed more explicitly upon the student, rather than upon the teacher or school system as it is now. Many topics of instruction and learning do not have to be directed by the teacher nor do they require that a teacher be present. It is likely that eventually tutor mode CAL will be a standard, or even dominant, mode of instruction/learning. Because of this and other benefits of tutor mode CAL discussed in this section, we should continue to expand usage of tutor mode CAL, even in situations where it is not yet 100% cost effective relative to conventional modes of instruction. By doing so we are investing in the future value of our students' education.

Some teachers fear that tutor mode computer assisted instruction will disrupt the school system and replace teachers. This will certainly not be true in the near future. The number of microcomputers currently being used in the United States precollege education system is enough to give each student one minute of computer use per day. A ten-fold or twenty-fold increase in computer assisted learning would still have only modest impact.

But by the year 2000, we may well have one microcomputer for every two students. A typical student may use computer assisted instruction materials for an hour or two per day. The computer, rather than textbooks and other print materials, may be the dominant mode of instruction. The potential impact upon teachers is not clear.

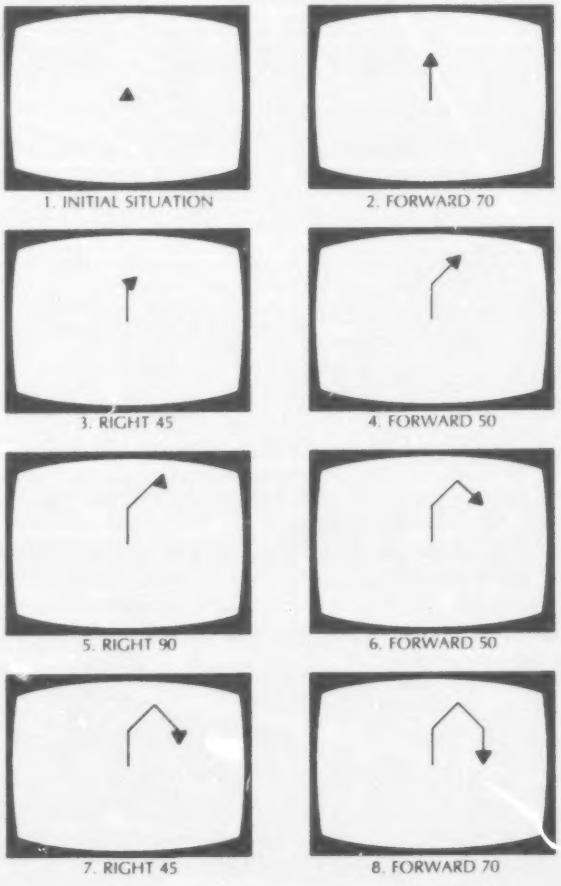
Computer Assisted Learning, Tutee Mode

In tutee mode CAL, a student acts upon a computer; the student is in charge, directing the interaction and learning by doing. The computer helps to provide a rich learning environment, but the computer is not preprogrammed with information to be taught to a student. Tutee mode CAL generally requires that a student learn quite a bit about a computer system and its language.

Reading provides a good analogy. A young student must expend considerable effort to master the rudiments of reading. Initially a student's aural and visual skills far exceed his/her reading skills in acquiring new information. But eventually reading skills increase and a new world opens— the printed record of the accumulated knowledge of the human race. Third graders may learn more about dinosaurs than their teachers know. A seventh grader may read about electricity, attaining a level of knowledge far beyond that of leading scientists two hundred years ago.

Similarly, students first encountering computers and a programming language in tutee mode CAL must focus upon learning the rudiments of a programming language. But eventually enough of the language is learned to open up new worlds for exploration and learning. If the computer and language system are appropriately designed, most students can move rapidly from an emphasis on the study of the computer system into an emphasis on learning other material.

The Logo language illustrates this quite well. Logo was specifically designed to be used in tutee mode CAL by elementary school students. Initial instruction may consist of learning to turn on the computer system and being shown a few simple examples. When the system is turned on, a pointed arrow called a "turtle" is displayed. This turtle draws lines as it moves about the screen following directions specified by the computer user.



Even very young children can quickly learn commands such as FORWARD, BACK, RIGHT and LEFT. FORWARD and BACK are accompanied by a distance while RIGHT and LEFT are accompanied by an angle measured in degrees. The commands have abbreviations FD, BK, RT and LT respectively.

Already the child is dealing with distances and angles— that is, with geometry. How can the turtle draw a house, a clown or a flower? After just a few

minutes of instruction, the focus changes from learning Logo to the solving of some problem.

As students progress, they will find a need for additional Logo language tools. Thus, students will repeatedly switch from working on a problem to learning more about the language and then back again to the problem.

In the above example the Logo system is used to create a geometry environment. This is a rich, deep environment; an entire secondary school geometry course has been embedded in this environment.[5]

A modern word processing system provides another example of tutee mode CAL. Such a system allows material to be typed, edited, stored and output. It also contains a spelling checker, which can help to identify misspelled words.

It requires some initial effort for a student to learn to use a word processing system. But the rewards are well worth the effort. Writing becomes more fun and correcting errors is no longer a major problem. A student can go through a number of drafts of a report or essay, trying out various ideas and continually improving the quality. The nice looking computer printout is a potent reward.

In the past, word processing has been quite expensive, so it has been used primarily in large business offices. Now, however, microcomputers have brought the cost of word processing within the reach of many millions of potential users. It is evident that most offices will eventually make use of this technology. As word processing comes into our educational system at all levels, the impact will likely prove to be dramatic. De-bugging, the systematic detection and correction of errors, will become a standard part of writing. Because the final product is nice to look at, more students will collect and display their writing. Perhaps we will spawn a new generation of writers!

The key idea of tutee mode CAL is using a computer system to create a new, rich, interesting learning environment. Tutee mode CAL can provide environments such as art, music, the physical sciences, and so on. In music, for example, we know that quite young children can develop a good ear for music and can learn to play musical instruments. With an appropriate computer-based environment, the same children can also compose music. The computer interacts with the composer, stores the composition and plays it when requested. The composer (the child) creates the musical composition, edits it and experiences the pleasure of being creative.

Art education provides another good example. There now exist excellent computer assisted "painting" programs. A student can paint a picture on a color television screen. Working with the computer system, the student can animate a picture, change the shape, size and color of its objects and experiment with perspective. Such experimentation is a powerful aid to learning art. Moreover, it provides a solid foundation for learning about computer graphics and for understanding how computers are used in the production of television and movies.

Tutee mode CAL can be done on any computer system. But obviously it will be more successful if the hardware and software are specifically designed to facilitate learning. An interactive Logo system is far superior to a batch processed COBOL system for young children. Eventually we will have a wide variety of computer hardware/software systems specifically designed to facilitate tutee mode CAL. In some disciplines it is likely that tutee mode CAL will prove to be a more superior aid to learning than conventional modes of instruction or tutor mode CAL. In the years to come, tutee mode CAL will certainly play an important role in education.

Tutee mode CAL is in its infancy. Since the research and experience bases are still quite modest in size, it is difficult to formulate precise student-oriented computer literacy goals in this area. Certainly all students should have an opportunity to explore a variety of learning environments based upon tutee mode CAL. Word processing, geometry, art and music learning environments are available through several different computer systems. These provide a good starting point for introducing students to the power and pleasure of using tutee mode CAL. As with tutor mode CAL, part of the goal is to help students learn about learning. Some students will find that tutee mode CAL is especially suited to their learning styles and academic interests.

The foundation of tutee mode CAL is discovery-based learning. A computer system is used to help create a rich educational environment and the student is encouraged to work in this environment. For many students, discovery-based learning is very effective—rapid and deep learning does occur.

But there are many other students who seem to flounder in a discovery-based learning environment. They need a more structured environment and more guidance from teachers.

The same two points can be made for teachers. Some teachers function well in a discovery-based learning/teaching environment. But many others have had little or no experience and instruction in discovery-based learning/teaching. For them, tutee mode CAL may be quite threatening.

II. An Aid to Problem Solving

Problem solving is a central and unifying theme in education. Any discipline can be framed as a hierarchic set of problems to be solved. Instruction in a discipline leads to understanding the nature of its problems: what problems have been solved and ways to solve some of these problems; what problems have not been solved and ways to formulate and attack new problems.

A key aspect of problem solving is building upon the work of others. This work is stored in a variety of ways. Each discipline has developed vocabulary, notation and tools specifically designed to aid in representing and solving its problems. The vocabulary of mathematics, music and psychotherapy are each highly

specialized, and a given word may have different meanings in these three disciplines. For example, a group in mathematics is not the same as a group in psychotherapy. The vocabulary in each discipline has been carefully developed to allow precise communication of important concepts in the field. Books, journals and other writings are "coded" in these vocabularies, and they constitute a standard way of storing the accumulated knowledge of a field. Still and motion pictures, video recordings and audio recordings are other modes of storage.

Much knowledge is stored in the form of machines that people learn to use: the telescope, microscope, telephone, television, clock, radio, automobile, and so on. A person can learn to use these machines to solve various problems without mastering all the details of constructing the machines or understanding how/why they work in the manner they do. Very young children can learn to use a telephone, television or record player.

The embodiment of information in a machine is a key idea. A young child can learn to read music and to play a musical instrument without learning the details of music theory or the design and construction of musical instruments. A child can learn to use a telephone system; a young adult can learn to drive a car. All of these activities involve building upon and using the work accomplished by others.

Computers, although they are also "merely" machines, provide a unique, new way to store knowledge. We can (roughly) divide the computer storage of knowledge into two categories—passive and active. Passive storage in a computer is analogous to storage in the form of books, films, etc. Computer storage may be more efficient perhaps, but not qualitatively different. Written materials can be stored on a magnetic disk or tape, and the material can be retrieved with the aid of a computerized information retrieval system. Similarly, pictures or sound can be digitized for computer storage and later reassembled for playback. This may be faster and more convenient than non-computerized methods, but in itself does not represent a profound change.

Active storage of information is illustrated by even the simplest four function calculator. The calculator "knows" how to add, subtract, multiply and divide. It stores this knowledge in an easily usable form. This sort of active storage of knowledge is akin to the storage of knowledge in other machines such as the telephone, microscope, alarm clock or stereo system.

Computers epitomize active storage; they are specifically designed for both the storage and retrieval of information, and for the manipulation of the stored information. A computer program can contain information on how to do something in a form that the computer can use to carry out the actual process. The computer can act upon the world, controlling industrial processes, routing telecommunications and helping to solve a wide variety of other problems. A computer can be a relatively smart machine, storing and using knowledge and skills beyond those of many of its users.

The active storage of information normally shortens the time it takes to manipulate or retrieve the information, as well as changing the knowledge and skills needed. Even the four-function calculator illustrates this. Grayson Wheatley estimates that a typical student spends two years of mathematics education instruction time in grades 1-9 mastering paper and pencil long division.[6] How radical is it to suggest that the time spent on this topic be halved and that students be given calculators?

The calculator example is particularly interesting because of its potential to greatly change arithmetic education. With some assistance from teachers, almost all first grade students can develop an intuitive understanding of addition, subtraction, multiplication and division. They can develop the ability to mentally solve simple examples of all four of these types of problems.

But progress in developing paper and pencil computational skills is slow. A typical student learns paper and pencil addition and subtraction by the end of the third grade and then moves on to multiplication and division. Consider the impact of giving all third graders calculators and allowing their unlimited use. The emphasis would be changed from computation to problem solving—from rote memory to higher level cognitive processes. Little research has been done upon this type of possible change to the curriculum.

And if calculators have the potential to make such a large change in the curriculum, what about computers? Eventually, computers will be nearly as common as calculators are today. What *should* students be able to do mentally? with pencil and paper? assisted by a calculator or computer? These are very difficult questions and will certainly challenge educators for decades to come.

Progress in hardware, software, artificial intelligence and computer assisted problem solving in all disciplines is continuously expanding the totality of problems that computers can solve or help to solve. The idea of a knowledge-based computer system is now well entrenched and growing in importance. What does a chemist, geologist, mathematician or physician know that a computer might be programmed to know? In these and many other disciplines, intense research efforts are producing computer systems that perform at an expert level. That is, computer systems can solve or help solve a variety of nonroutine problems complex enough to challenge a human expert. The number of these expert-level knowledge-based computer systems will grow rapidly over the next ten to twenty years. Thus, for any particular problem area that a student might study, it is likely that computers are already a very important aid in problem solving and that the importance of computers in that area is growing.

The computer literate student understands and uses computers as an aid to problem solving. This means that the student has studied problem solving and a variety of aids to problem solving. The student has used computers as an aid to problem solving over a period of many years in a wide variety of disciplines and

understands their capabilities and limitations. Given a problem, the truly competent student can decide if a computer is a useful aid compared to other aids/approaches to solving the problem.

If a computer is to be used to help solve a problem, appropriate software is necessary. Previously written programs (often called canned programs, library programs or packaged programs) are readily available for many general types of problems. Some of these library programs are easy to use and easy to learn how to use. Others require substantial instruction and practice. Indeed, learning to use certain packaged programs is roughly equivalent in difficulty to learning a general purpose programming language. There is no clear dividing line between programming and using problem-oriented packages of library programs.

In many situations an appropriate library program is not available. An existing program may need to be modified, several pieces of existing programs may need to be combined or a new program may need to be written. Thus, instruction in computer programming surfaces once again as an important part of computer literacy. We will discuss this in more detail later.

If the use of computers as an aid to problem solving is taught and integrated into the curriculum, some parts of the curriculum will substantially change. The greatest changes will be in areas where we know a lot about problem solving such as in mathematics and the sciences. But our curriculum contains a number of other areas in which a computer can solve problems or can be of substantial assistance in solving them. We will need to decide what we want students to learn to do by "conventional" methods and to what extent "knowing" an area includes knowing how to make use of a computer to solve problems in this area. Students need to know which aspects of the problems they are studying can be handled by a computer.

The ideas of computer literacy raised in this section are dependent upon the capabilities of computer hardware and software and so will change over time. As with CAL, students should become familiar with the best of modern hardware and software, since continued rapid progress is to be expected in both. This type of computer literacy is multidisciplinary. Its proper achievement requires that teachers be computer literate with respect to their own disciplines. The Association for Computing Machinery has made recommendations on teacher education.[7]

Most teachers today are not computer literate within their own teaching areas. They do not know how computers can help solve the problems of their disciplines. Moreover, most schools of education are not yet producing computer literate graduates. For the next decade or two our educational system faces a serious problem. Computer systems will become increasingly capable aids to problem solving, while the computer knowledge of most educators will continue to lag far behind. It will take a distinct effort on the part of our educational system to significantly improve this situation.

The teacher education problem is being attacked from two directions. First, many school districts now have computers-in-education committees that work to set student-oriented goals and to develop the needed teacher inservice courses. Second, colleges of education are beginning to put significant effort into both preservice and inservice computer education. A recent collection of eighteen papers discusses what various colleges of education are doing and gives recommendations for this phase of teacher education.[13]

III. An Object of Learning in Itself: Discipline of Computer and Information Science

Computer and information science is a new and important discipline. It is now well established in most major colleges and universities and is rapidly growing in stature. In the United States alone there are nearly 400 bachelor's degree programs and nearly 100 doctoral programs in this field. Hundreds of journals devote all or part of their content to computer and information science topics, and the research journals of almost every other discipline occasionally carry computer-related articles.

Computer programming is one part of computer and information science, and learning some programming is an essential step in understanding computer and information science. We are talking about a student-oriented, non-professional level of computer programming knowledge and skills. A student should be able to program well enough to be able to attack the types of problems being studied in the school curriculum and to make effective use of the tools being taught. When the topic being studied is part of computer and information science, it is even more important that students write programs.

Computer programming involves learning a language. More importantly, it involves developing and practicing problem-solving skills. Top down analysis, segmentation, testing and debugging are fundamental ideas best learned through hands-on experience. These programming-related ideas carry over to problem solving in many other academic areas.

Thus, we are led to include computer programming as part of computer literacy via our analysis of tutee mode CAL, through our analysis of problem solving, and also through the importance of computer and information science. But none of these gives a precise statement of the level or nature of programming skill appropriate to computer literacy.

To specify goals for computer programming instruction more precisely, we need to define what it means to program. Many interactive computer systems function in both an immediate execution and a delayed execution mode. In the immediate execution mode, statements such as FORWARD 50 from Logo or PRINT 72/389 from BASIC are immediately carried out by the computer. These statements can be thought of as one-line programs. A student who writes such statements is programming. Even this level of programming skill is

useful in learning geometry, arithmetic or more about programming. In the delayed execution mode, a student prepares a sequence of one or more statements and enters it into a computer system. After entering the sequence of statements, the student can direct the computer to execute the program.

Word processing and information retrieval also provide examples of computer systems that have both immediate execution commands and delayed execution capabilities. Learning to use a word processing system or an information retrieval system is learning to program. It is true that one is learning a special-purpose language, rather than a general-purpose language such as BASIC, Logo or Pascal. However, the same general principles apply, and it is clear that the computer user is directing the system.

In both the immediate execution and the delayed execution modes, a student is making use of very sophisticated software. In the early years of computers, most available software was written in a general-purpose language. However, there has been a strong historical trend towards special-purpose software systems. This became evident quite early in statistics. For more than 20 years now, there have been quite sophisticated statistical program packages. Learning to use such a system is comparable in difficulty to learning a general-purpose programming language.

One of the most successful and widely used applications systems for microcomputers is known as an electronic spread sheet. In essence, the computer display becomes a large two-dimensional table. Columns in the table may be related to each other by simple formulas. Then, changes to the figures in one column cause automatic updates to figures in other columns. The electronic spread sheet is such a useful tool to many business people that they purchase a computer system just for this particular application. It is quite appropriate that business-oriented secondary school students learn to use such systems.

Including the use of information retrieval, word processing, statistics and other applications systems in our definition of programming greatly broadens the scope of what it means to learn to program. Learning to use applications systems is a rapidly growing part of computer programming; it will eventually be the dominant part. All students can learn to use applications systems, and this is an essential part of becoming computer literate.

In having students learn to program, the goal is to attain a functional level of knowledge and skill—a level useful in studying and attacking problems in every academic area. The analogy with reading and writing is again useful. For a first grade student, reading and writing are specific academic disciplines; it takes considerable effort to learn to read and write. But a high school student uses reading and writing as tools to study other disciplines.

Our educational system has had many hundreds of years of experience in helping students learn to read and write. We know that most students can develop a functional level of reading and writing literacy. We know that instruction can begin in the first grade or earlier and that the rate of progress towards functional literacy varies considerably among students. Moreover, we recognize there is a significant difference between a functional literacy level and a professional level. Some students study journalism, writing and literature in college or graduate school. They develop much higher levels of skill and knowledge in reading and writing than the general populace.

Our educational system has only limited experience in helping children learn to program a computer. But we know that if appropriate computer facilities and teacher knowledge are available, then elementary school students can learn to program. A child's initial exploration of Logo can be a valuable learning experience. The experience rapidly gets into problem areas such as geometry where the computer system and the student's programming skills become useful aids in learning new non-computer material.

The use of word processing at the elementary school level is in its infancy. Seymour Papert's work has shown that even learning disabled children can learn to use a word processing system. There is some evidence that success in using word processing carries over to other academic areas, leading to an overall improvement in academic performance.

Given adequate time, computer access and instruction, most middle school and junior high school students can learn to program in a language such as BASIC, Logo or Pascal. Such students can learn to use an information retrieval system, a word processing system and other applications systems. Currently, the great majority of computer programming instruction at the precollege level focuses upon general-purpose language systems, especially BASIC. This emphasis will gradually shift as students and educators come to appreciate the value and power of the applications systems.

The key to functional computer literacy is having a supportive environment in which students can continue to use the computer knowledge and skills they are acquiring. A seventh grade student can learn to use a word processing system and an information retrieval system. These are general-purpose tools—the student can use them in almost all academic areas. Skill in using these systems will grow as the student grows in overall academic accomplishment, provided adequate computer access and encouragement are available.

Another example is provided by computer graphics systems. A graphics system makes it possible for a person to easily draw a bar graph, pie chart, scatter plot or function graph. Drawing graphs is useful in social sciences, physical sciences and mathematics. Initial exposure to a comprehensive graphics system might occur in the ninth grade. Subsequently, students could use this system in a variety of courses for the remainder of their educational careers.

To be more specific, consider a student progressing through the typical algebra, geometry and second year

algebra sequence of courses. Computer graphics is a useful tool in all of these courses, both as an aid to problem solving and as an aid to understanding the topics being studied. Graphical representations of functions, for example, can help to improve one's intuitive insights into functions and their uses. A computer literate student taking these math courses would understand uses of computer graphics in the courses and would make frequent use of this important tool.

Along with instruction in special-purpose and general-purpose computer programming systems should come instruction in computer science. Introductory ideas can be woven into the curriculum at all academic levels. A formal computer science course might be given at the high school level.

The Association for Computing Machinery (ACM), working through its Elementary and Secondary Schools Subcommittee, has developed a year-long computer science course for high school students.[8] The course is intended to be roughly comparable to high school biology in its difficulty, and the hope is that eventually it will have a similarly wide audience. The course has a relatively low mathematics prerequisite. Its content is balanced between computer programming, problem solving and a variety of topics from computer and information science. A detailed, week by week outline for such a high school computer science course is given in Jean Rogers' booklet, *An Introduction to Computing: Content for a High School Course*, published by ICCE.[9]

Computer science includes topics such as artificial intelligence, business data processing, computers and society, computer graphics, information retrieval, and modeling and simulation. It also covers the design, representation, testing and debugging of procedures to solve problems. These latter ideas carry over to other (non-computer) academic areas, providing students with some general-purpose problem-solving skills.

These skills and their underlying ideas are quite useful and powerful. Consider debugging. Currently, most students are taught that the math work they do is either "right" or "wrong." They do not explicitly learn that their "wrong" work may be mostly correct and merely need some debugging. Contrast this with learning to write. The idea is well accepted there that a student's work may need debugging. Teaching the idea of bugs and debugging could profoundly change mathematics education at the pre-college level.

It is perhaps too early to say that a high school level computer and information science course is an essential part of computer literacy. But already we can see movement in that direction on the part of some colleges and universities. That is, it seems likely that ten years from now many colleges and universities will place entering freshmen into a remedial computer literacy course if they have not acquired such knowledge previously.

It is also difficult to know what employers will expect of students entering the job market ten years from now. The rapid proliferation of computers suggests that quite a high level of computer literacy will be expected. The ACM course might become part of a definition of the expected level of computer literacy.

The standards for computer literacy discussed in this section tend to come from higher-level authority (for example, the ACM or ICCE), rather than being apparent to the student. Not all students will easily accept that an ACM or ICCE-recommended body of knowledge and specific skills will be useful on the job or in college. Moreover, we cannot say with certainty that such a course is indeed appropriate. For many years to come, people will be able to acquire needed levels of computer-oriented skills on the job or in their higher education programs. But precollege students who have acquired this level of computer literacy will have a distinct advantage in seeking jobs and/or in continuing their formal education.

Although this booklet focuses mainly on precollege computer literacy, the emerging pattern of college level computer literacy provides useful information. In the past few years enrollment in college computer science and computer literacy courses has doubled and then doubled again. College students are aware of the value of having a solid functional level of computer science knowledge. It is likely that this awareness will spread to high school students, leading to a rapid growth in demand for computer-related courses. Most colleges are hard pressed to meet the demand, and the same problem is likely to occur in many high schools.

IV. Entertainment

Computers are a rapidly growing form of entertainment. They compete successfully with television, stereos, books and movies. They are quite important in the lives of many students and can have either a negative or a positive effect. We should remember that the typical eighteen-year-old in the United States has watched more hours of TV than s/he has attended school! Twenty years from now we may be making similar statements about student use of computerized entertainment systems.

As with CAL, use of computers as a form of entertainment can be divided into two main categories. The designing and implementing of programs is fun for some students. If the program plays a game or simulates an alien environment, it is especially fun. Some people spend a significant percentage of their leisure time writing, testing and improving such programs. They often develop a very high level of programming skill, a level which generally exceeds the skill most students develop through programming courses offered in schools.

But the great majority of computer use for entertainment is game playing. Some computerized games can be learned and perhaps even mastered in a matter of minutes; however, there is a growing collection of computerized games requiring dozens or even hundreds of hours of effort to master. Extensive learning or the development of a high level of hand/eye coordination is needed.

Typical of the sophisticated computer games are the computerized variations of Dungeons and Dragons. These are fantasy games in which one explores multi-leveled dungeons, searching for treasures and fighting dragons and other creatures. The games are quite complex and playing them well requires a good memory, good attention to detail and concentration. While careful studies of their educational value have not been done, it seems evident that such games are intellectual in nature, and thus have educational value. Who is to say that learning to play chess is a better use of one's time than learning to play a computerized fantasy game?

Quite good computer programs now exist to aid musical composition or ear training, and computerized aids for artistic creation are also available. It is not difficult to include these in the realm of entertainment, but they also have clear educational value.

A computer program to play chess, checkers, backgammon or Othello can be a challenging opponent and an excellent aid to learning one of these games. Many other problem-solving situations can be formulated as interesting games, involving both entertainment and learning.

There is no clear dividing line between entertainment and education. Indeed, if learning is fun, more and better learning tends to occur. Thus, students should be given the opportunity to make use of CAL materials that are both educational and fun. They should learn to be critical of learning aids that are unnecessarily dull.

There appears to be little need to give students instruction in how to use a computer as a form of entertainment. Students quickly learn this on their own or from their peers. But the study of entertainment, or more appropriately the study of leisure time, is now considered to be important in modern education.

A computer literate student has experienced the use of computers as a form of entertainment in a variety of situations. The student has studied various forms of leisure time activity and how computers fit into this field. The student has made a conscious and reasoned decision as to the role computerized entertainment will play in his/her life at the current time.

V. A Part of Their Future

Students in junior high and high school often are actively interested in the social problems of our society. They study these problems, and they begin to work toward helping solve the problems. While computers are useful aids to problem solving, they are also a new source of problems. For example, computers make possible very large, easily accessible data banks. Such data banks may contain detailed records on a person's schooling, criminal history, federal and state taxes, medical history and employment. The 1984 "big brother is watching" era is nearly upon us.

A computer literate student has studied the role of computers and privacy. The student is knowledgeable about the capabilities and limitations of computerized systems that store data about people and their activities. The student is able to function as a well-informed citizen in helping to preserve individual freedoms and those aspects of individual privacy that are so important in our society.

Computers represent change, and computers are a change agent. It is generally agreed that one major goal of education is to help students prepare to cope with situations they will encounter later in life. Every student will encounter new and different situations; every student will encounter change.

Many of these changes will be based upon developments in science and technology. We can expect continued rapid progress in such diverse fields as medicine, genetic engineering, telecommunications and automation.

At the heart of scientific and technological progress is the accumulation and application of knowledge. And it is here that computers are making a substantial contribution. Computers, supported by the general knowledge being developed through the field of computer and information science, have become an indispensable part of our science and technology.

Moreover, computers are one of the most rapidly changing parts of science and technology. The rapid progress in computer hardware, software and applications that we have seen in the past thirty years seems destined to continue for the next thirty. These past thirty years have taken us from the UNIVAC I, costing well over a million 1951 dollars, to the portable and/or handheld microcomputers of today. Many of these microcomputers exceed the UNIVAC I in capability, while costing less than one-thousandth as much!

It is fun to make a conjecture about what the next thirty years will bring. The Dynabook project and Smalltalk-80 language based on Alan Kay's ideas are especially exciting.[10] Work began at Xerox Corporation in the early 1970s on a handheld computer that would have a high resolution graphics display screen and a very powerful, modern language. Preliminary versions of the Smalltalk language were extensively tested with children, although the overall development project is now aimed mainly at other markets.

Thirty years from now we can expect to have inexpensive handheld computers that exceed today's million-dollar machines in capability and ease of use. Computers will be more common than television sets are today. There will be large libraries of programs that can be used to help solve a continually expanding range of problems. All educated people will make everyday use of these computer libraries.

It is important that students understand the rapid changes that are occurring in the computer field and what the future is apt to bring. In particular, how will computers affect the job market and the types of jobs that are available? Current estimates are that computer-based automation of manufacturing in the United States will eliminate ten million jobs over the next twenty years. The office of the future will utilize word processing, computerized information retrieval and

electronic mail. Knowledge and skills needed to function in this office environment of the future are different from the knowledge and skills that most students are acquiring in today's schools. The publishing and advertising industries will be drastically changed by computerized video disks and computerized information retrieval systems in people's homes. The postal system will be substantially changed by electronic mail.

A student who understands these potential changes can plan accordingly. Decisions on education and career goals should take into consideration how computers are changing our world. This is an important part of computer literacy.

FINAL REMARKS

One common way to talk about computer literacy is to discuss knowledge, attitudes and skills. Various instruments have been developed to test these aspects of computer literacy, and test scores serve to define levels of competency. Another approach is to specify course goals and objectives and to develop specific course content to implement these goals, such as Neill and Ricketts have done.[11,12]

In late 1982 the U.S. Federal government made a grant to the Educational Testing Service, Princeton, New Jersey to work on developing a definition of computer literacy and an instrument to measure literacy. Part of the work on the project has been subcontracted to the Human Resources Research Organization of Alexandria, Virginia. The idea is to do a national study of school superintendents, principals, teachers and students to measure their levels of computer literacy.

The approach being taken is based upon the work of Neill and Ricketts.[11,12] It will include a multiple item test plus the gathering of some information on how these groups of people actually use computers. Likely the computer literacy measurement instruments will be ready for initial testing in the fall of 1983.

We have not attempted to use these approaches here, nor have we discussed their merits. Rather, we have used a different approach, based on the idea that a computer can have a personal impact upon the student, and that the student will be self-motivated to acquire a certain level of computer literacy because of the personal value of computers. This assumes, of course, that appropriate learning opportunities are made available to students. Students need easy, everyday access to computers if the personal computing ideas of this paper are to be implemented successfully. Moreover, students need computer literate teachers.

The conclusion reached in this booklet is that computer literacy is a functional knowledge of computers and their effects on students and on the rest of our society. This knowledge should be at a level compatible with other knowledge and skills a student is acquiring in school. It is a knowledge based on understanding how computers can help us learn, how computers can help us solve problems, what computer knowledge is essential to a modern understanding of other academic areas, what is included in the field of computer and information science, computers as entertainment and what role computers will play in our changing world. This approach to computer literacy changes easily as computers become more readily available and easier to use, as we learn more about computers and integrate the knowledge into the curriculum, and as the use of computers becomes commonplace in homes, businesses, government and schools.

If students are to acquire a functional knowledge of computers, our school system will need to provide substantial computer equipment and instruction. New courses will need to be developed and many current courses will need to be revised. Support materials such as lesson plans, student workbooks, textbooks, films and other computer-oriented materials will need to be developed. Teachers will need to develop their own computer literacy.

The problem is large, but the goal is clear. Functional computer literacy is important for all students.

REFERENCES

1. Mildred Beavers, Eugene Collins, Ed Herber, Dean Larsen: A Course in Algebra and Trigonometry with Computer Programming, The University of Colorado, 1969.
2. Conference Board of the Mathematical Sciences Committee on Computer Education: *Recommendations Regarding Computers in High School Education*, April 1972.
3. Robert Taylor, Editor: *The Computer in the School: Tutor, Tool, Tutee*, Teachers College Press, 1980. This book contains three papers by Alfred Bork, four papers by Thomas Dwyer, four papers by Arthur Luehrmann, four papers by Seymour Papert and four papers by Patrick Suppes.
4. Seymour Papert: *Mindstorms: Children, Computers and Powerful Ideas*, Basic Books, New York, NY, 1980.
5. Harold Abelson and Andrea diSessa: *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*, The MIT Press, 1981.
6. Grayson Wheatley: "Calculators in the Classroom: A Proposal for Curriculum Change," *Arithmetic Teacher*, December 1980.
7. Taylor, Poirot, and Powell: "Computing Competencies for School Teachers," Association for Computing Machinery joint SIGCSE, SIGCUE "Topics" issue, ACM Order No. 812810, January 1981.
8. Jean Rogers and Richard Austing: "Recommendations for a One Year Secondary School Computer Science Course," Association for Computing Machinery joint SIGCSE, SIGCUE "Topics" issue, ACM Order No. 812810, January 1981.
9. Jean Rogers: An Introduction to Computing: Content for a High School Course, International Council for Computers in Education, University of Oregon, Eugene, Oregon, 1981.
10. *Byte Magazine*, Volume 6, Number 8, August 1981. This issue contains thirteen articles about the Smalltalk language and its potential applications.
11. Michael Neill: *An Empirical Method of Identifying Instructional Objectives for a High School Computer Literacy Curriculum*, Ph.D. Thesis at the University of Oregon, June 1976.
12. Dick Ricketts (Director of Computer Education Goal Development): *Course Goals in Computer Education, K-12*, disseminated by the Northwest Regional Educational Laboratory, Portland, Oregon, 1979.
13. *Topics: Computer Education for Colleges of Education*, Association for Computing Machinery, New York, 1983, available from ICCE, 135 Education, University of Oregon, Eugene, Or 97403.
14. *The Computing Teacher*, Vol. 10 #3, Special Logo Issue, University of Oregon, Eugene, OR 97403.

ICCE PUBLICATIONS

You should know about the 11 computer education books and booklets we publish. Books about computers and you—about computers and your students—about computers and special applications. There's a lot to learn about computers and education, and ICCE booklets can help you learn it!

• **The Evaluator's Guide to Microcomputer-Based Instructional Packages** can help you make informed, intelligent software purchasing decisions. This recently revised step-by-step guide to concise software evaluation—with sample reviews, evaluation forms and Glossary—is an invaluable tool in effective software review and selection. $2.50, 48 pages.

• **An Introduction to Computing: Content for a High School Course** proposes a unit-by-unit course outline for computer science instruction at the secondary school level. Recently revised, and designed for a year-long course, the booklet contains a variety of activities emphasizing applications, programming, computer environment and social impact. $2.50, 48 pages.

• The **School Administrator's Introduction to Instructional Use of Computers**, a good fundamental overview for school administrators and board members, maps out the basic elements of computer education and some ways that computers are affecting today's curriculum. Written in four parts the booklet introduces the subject, reviews instructional uses of computers, looks at the problems and processes of goal development, and closes with a Glossary and brief guide to appropriate periodical literature. $3.00, 64 pages.

• The **Teacher's Guide to Computers in the Elementary School** is designed for the teacher with little practical computer experience. The booklet begins with a brief preface followed by an introduction to the subject of computers. The second section talks in more detail about computers and education. The third section discusses specific activities, many of which do not require a computer. Finally, there is a Glossary and a look at some additional resources. $2.50, 48 pages.

• **Precollege Computer Literacy: A Personal Computing Approach** talks about the need for professionals in the field of computer education to recognize and implement a universal standard of precollege computer literacy. Via an analysis of personal computing and those aspects of computers that have direct impact on students, the booklet briefly discusses and defines computer literacy goals for elementary and secondary schools. $1.50, 28 pages.

• An **Introduction to Computers in Education for Elementary and Middle School Teachers** is considerably more explicit than the *Teacher's Guide*. Designed for self-instruction, inservice or preservice use, the text is suitable for teachers at all levels. Substantially larger than most of the ICCE booklets, this in-depth work includes eight chapters with over 75 applications and exercises (many of which do not require computer access) that can be used at the elementary and middle school levels. $7.00, 94 pages, 8½x11.

• The **Parent's Guide to Computers in Education**, one of our most recent publications, is a definitive resource for parents who wish to become actively involved in their children's computer education experiences. The booklet talks about why children should use and learn about computers, and what parents can do to help. There is a section on some of the ways students and teachers are using computers in the classroom, and suggestions for obtaining hands-on experience outside the classroom for parents and children. Checklists are included for parents planning to purchase a computer for the home. The closing section contains a Glossary and list of resources. To put the subject into perspective, a lighthearted piece by Merle Marsh, "Comes the Dawn (If Only I Can Find the Switch)", runs throughout the booklet. $3.50, 80 pages.

• **Learning Disabled Students and Computers: A Teacher's Guide Book** was written to help those teachers who are responsible for meeting the needs of children legally identified as learning disabled. Discussing software and hardware, the booklet also emphasizes educational computing theory, concerns and a vision of computers in education for learning disabled students. The Resource section contains a Glossary and references to relevant books, publishers and organizations. $2.50, 48 pages.

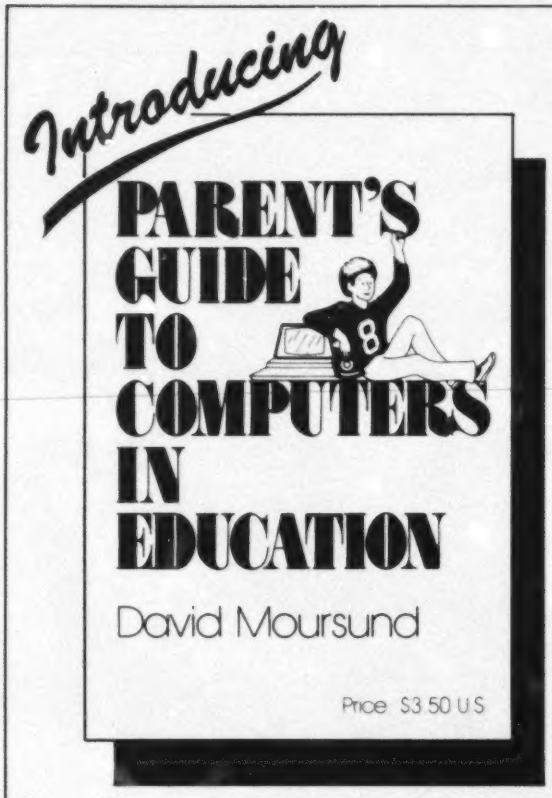
• **Computer Technology for the Handicapped in Special Education & Rehabilitation: A Resource Guide** is a detailed, up-to-date annotated bibliography of resources in the area of computers for the handicapped in rehabilitation and special education. Containing an author index, subject index and referenced quotations, the focus of the guide is a body of nearly 200 detailed annotations of current writings in the field. Specific subject areas include communication, computer assisted instruction, disability/handicap, functional aids, microcomputers/applications, newsletter/bulletins, periodicals, rehabilitation, special education, teachers/service providers, universities/colleges and several others. Published in January of 1983, this is one of the most current subject source books available. $7.00, 94 pages, 8½x11.

• **Microcomputers in the Classroom—Dreams and Realities** is directed to educators, administrators, curriculum committees, and other professionals concerned about the practice and potential of computer education. Written by Dr. Henry Jay Becker of The Johns Hopkins University, this booklet provides an overview of the existing computer education environment as it relates to school systems and policy. Dr. Becker has included a summary of current theory, a review of relevant research and an outline of some major factors that will influence purchasing, implementation and goal-setting decisions. Divided into fourteen sections such as "management of instruction" and "curriculum-modifying applications of microcomputers," the booklet concludes with suggestions for improved performance on the part of researchers, developers and school systems. $3.50, 64 pages.

• **Computer Literacy Activities for Elementary and Middle School Students**, designed for teachers who are just getting started with computers, contains eleven activities which can be used at a variety of grade levels. Many can be easily integrated into the existing curriculum and most do not require computer access. Eight authors combined their computer education knowledge and commitment to the field in producing this detailed work which includes a Glossary and Resources section. $3.00.

To purchase these booklets write to ICCE, Dept. CTA, 135 Education, University of Oregon, Eugene, OR 97403 or phone 503/686-4414.

Also from ICCE . . .



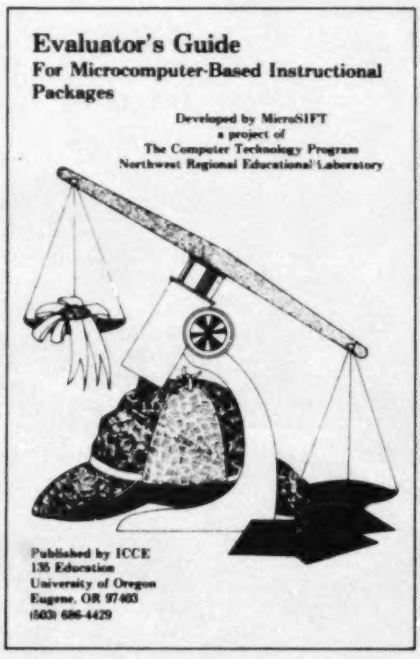
You spend a lot of time teaching students about computers. Shouldn't parents be learning something too? By reading the **Parent's Guide to Computers in Education**, parents can learn what is being done, and what needs to be done, to make their children computer literate. You need to tell them that the guide exists.

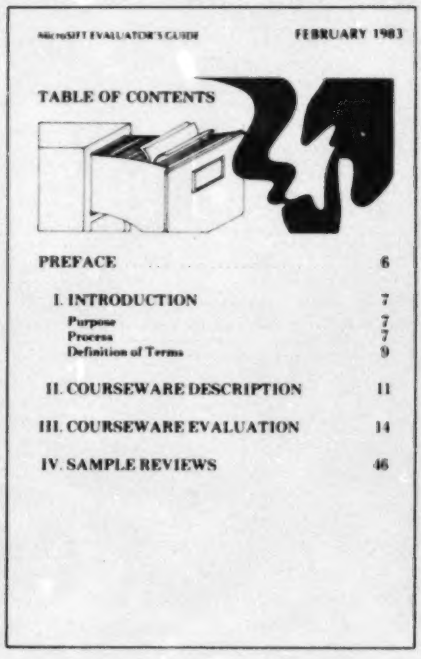
The **Parent's Guide to Computers in Education** is not dry academic reading. It treats the subject of computer literacy with sensitivity, insight and humor.

The booklet talks about why children should use and learn about computers, and what parents can do to help. There is a section on some of the ways students and teachers are using computers in the classroom, and suggestions for obtaining hands-on experience for parents and children outside of the classroom. For parents planning to purchase a computer for the home, there are Needs Assessment and Implementations Assessment checklists. The booklet also contains a Glossary and list of references.

A delightful and encouraging article by Merle Marsh, "Comes the Dawn (If Only I Can Find the Switch)," the story of one parent's journey into the computer age, runs throughout the work.

The booklet, which is 80 pages long, is available for $3.50 pre-paid, or by invoice ($2.50 will be added for handling) from the International Council for Computers in Education, 135 Education, Dept. PGI, University of Oregon, Eugene, Oregon 97403. Visa and Mastercard accepted.





Look no further for a comprehensive instructional software evaluation tool. *MicroSIFT Evaluator's Guide*, developed by the Computer Technology Program of the Northwest Regional Laboratory, is a substantial guide for evaluating instructional software. Contents include a summary of MicroSIFT's evaluation process, glossary, detailed description of all review components, review forms and sample reviews. (Revised edition available February 1983.)

QUANTITY	PRICE (U.S. FUNDS)
1-4 copies	$2.50 each
5-9 copies	$2.25 each
10-29 copies	$2.00 each
30-49 copies	$1.75 each
50+ copies	$1.50 each

VISA